

**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE QUITO**

**CARRERA:**  
**INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:**  
**INGENIERAS ELECTRÓNICAS**

**TEMA:**  
**DESARROLLO DE UN PROTOTIPO PARA APRENDIZAJE DE**  
**LENGUAJE BRAILLE ASISTIDO POR AUDIO**

**AUTORAS:**  
**PAOLA CAROLINA CABRERA CALDERÓN**  
**JENNYFER DANIELA MEJÍA ÁVILA**

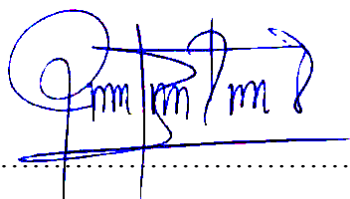
**TUTOR:**  
**GUSTAVO JAVIER CAIZA GUANOCHANGA**

**Quito, marzo de 2021**

## **CESIÓN DE DERECHOS DE AUTOR**

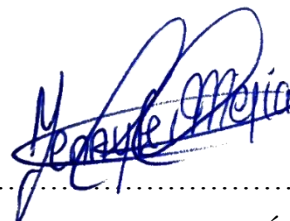
Nosotras, Paola Carolina Cabrera Calderón con documento de identificación N°1721040713 y Jennyfer Daniela Mejía Ávila con documentos de identificación N°1725405359, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: **DESARROLLO DE UN PROTOTIPO PARA APRENDIZAJE DE LENGUAJE BRAILLE ASISTIDO POR AUDIO**, mismo que ha sido desarrollado para optar por el título de: Ingenieras Electrónicas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autoras nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



Paola Carolina Cabrera Calderón

C.I.1721040713



Jennyfer Daniela Mejía Ávila

C.I.1725405359

Quito, marzo de 2021

## **DECLARACIÓN DE COUTORIA DEL DOCENTE TUTOR**

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico Académico, DESARROLLO DE UN PROTOTIPO PARA APRENDIZAJE DE LENGUAJE BRAILLE ASISTIDO POR AUDIO, realizado por Paola Carolina Cabrera Calderón y Jennyfer Daniela Mejía Ávila, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, marzo de 2021.



.....

Gustavo Javier Caiza Guanochanga

C.I.1721192191

## **DEDICATORIAS**

Dedico este trabajo de grado, primeramente a Dios por permitirme llegar a este momento tan especial en mi vida, por los triunfos y los momentos difíciles dándome sabiduría e inteligencia necesaria para cumplir con una más de mis metas, a mi mamá por ser mi apoyo a lo largo de mi carrera universitaria y a lo largo de mi vida, a mi papá por los valores que me ha inculcado y por su amor incondicional, a hermana por ser mi apoyo quien fue el principal cimiento para alcanzar mi meta tan deseada y a todas las personas especiales que me acompañaron en esta etapa, aportando a mi formación tanto profesional y como de ser humano.

Jennyfer Daniela Mejía Ávila

A las personas más importantes en mi vida, mis padres; por su cariño, esfuerzo y constancia; quienes fueron y siguen siendo mi apoyo día tras día.

Paola Carolina Cabrera Calderón

## **AGRADECIMIENTOS**

Gracias a Dios por permitirme el haber llegado hasta este momento tan importante de mi formación profesional, por tu amor y tu bondad que no tiene fin.

Gracias a mis padres por la vida que tengo, por los valores inculcados y los sacrificios que hicieron por mí, gracias a mi madre Luz Ávila por estar dispuesta a acompañarme en cada larga y agotadora noche de estudio en las que su compañía me animaban para seguir esforzándome, con sus palabras de aliento, gracias a mi difunto padre Donato Mejía por cada consejo y por cada una de sus palabras que me guiaran durante mi vida, porque siempre lo recordare.

Gracias a mis hermanos por sus consejos y palabras de aliento, especialmente gracias a mi hermana Lourdes Mejía brindarme un apoyo incondicional, permitirme estudiar y compartir en los momentos difíciles.

Gracias a la Universidad Politécnica Salesiana por permitirme continuar con mis estudios y convertirme en una profesional, a los profesores que nos impartieron sus enseñanzas y nos colaboraron en nuestro aprendizaje.

Finalmente, a todas las personas que me apoyaron y creyeron en mí muchas gracias.

Jennyfer Daniela Mejía Ávila

A mis padres, por toda esa sabiduría y humildad, por inculcarme principios y valores, por todo el esfuerzo y dedicación.

A mis padrinos, por asumir su rol y ser una guía en mi camino, por su cariño, motivación y apoyo.

A mi tío Luis Cabrera, quién con su carisma, sabiduría y liderazgo formó una familia muy unida, físicamente ya no está, pero sus enseñanzas y valores nos han ayudado a ser mejores cada día.

A los docentes de la UPS, a aquellos que se preocupan por que sus alumnos sean mejores no solo a nivel académico sino también personal, por su esfuerzo, paciencia y dedicación, por demostrar sus valores y representar una Figura a seguir.

Y a todos quienes han sido apoyo en los momentos difíciles y después de mis caídas me dieron la mano para levantarme.

Infinitamente gracias.

Paola Carolina Cabrera Calderón

## ÍNDICE GENERAL

<b>CESIÓN DE DERECHOS DE AUTOR .....</b>	<b>i</b>
<b>DECLARACIÓN DE COUTORIA DEL DOCENTE TUTOR.....</b>	<b>ii</b>
<b>DEDICATORIAS.....</b>	<b>iii</b>
<b>AGRADECIMIENTOS .....</b>	<b>iv</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>viii</b>
<b>INDICE DE TABLAS.....</b>	<b>x</b>
<b>RESUMEN.....</b>	<b>xi</b>
<b>ABSTRACT .....</b>	<b>xii</b>
<b>INTRODUCCIÓN .....</b>	<b>xiii</b>
<b>CAPÍTULO 1.....</b>	<b>1</b>
<b>ANTECEDENTES .....</b>	<b>1</b>
1.1. Planteamiento del Problema .....	1
1.2. Justificación.....	1
1.3. Objetivos .....	2
1.3.1. Objetivo General .....	2
1.3.2. Objetivos Específicos.....	2
<b>CAPÍTULO 2.....</b>	<b>3</b>
<b>FUNDAMENTACIÓN TEÓRICA.....</b>	<b>3</b>
2.1. Discapacidad visual .....	3
2.2. Discapacidad visual en el Ecuador .....	3
2.3 Lenguaje braille .....	4
2.3.1 Signo generador .....	4
2.4 Métodos de enseñanza y aprendizaje en niños .....	5
2.5 Teclado braille .....	5
2.6 Alfabeto braille.....	6
2.7 Formaciones especiales del Sistema braille .....	7

2.7.1	Letras mayúsculas .....	7
2.7.2	Números .....	7
2.7.3	Vocales acentuadas .....	7
2.8	Grados del sistema braille .....	8
2.8.1	Grado Uno .....	8
2.8.2	Grado Dos .....	8
2.8.3	Grado Tres.....	8
2.9	Parámetros dimensionales de los caracteres del Sistema braille .....	8
2.10	Alexa voice service .....	9
2.11	Componentes de AWS.....	10
2.11.1	AWS IoT .....	10
2.11.2	AWS Lambda.....	10
2.11.3	AWS EC2.....	11
2.11.4	Alexa Skill.....	11
2.12	MQTT .....	11
2.12.1	Características de MQTT .....	11
2.13	JSON.....	12
2.14	Node.js .....	12
2.15	Raspberry Pi modelo B + .....	12
2.16	Solenoides .....	13
2.17	Módulo conversor DC/DC.....	13
<b>CAPÍTULO 3.....</b>		<b>15</b>
<b>DISEÑO E IMPLEMETACIÓN .....</b>		<b>15</b>
3.1	Diagrama de bloques .....	15
3.2	Funcionamiento lógico del dispositivo.....	16
3.3	Instalación y preparación de Raspberry Pi .....	17
3.4	Instalación del Servidor de voz Alexa en la Raspberry Pi .....	17

3.5	Configuración AWS IoT .....	18
3.6	Creación de la habilidad personalizada de Alexa.....	20
3.7	AWS Lambda .....	23
3.7.1	Etapa de aprendizaje .....	26
3.7.2	Etapa de evaluación.....	28
3.8	Código Python .....	28
3.8.1	Código Python para el Teclado braille.....	28
3.8.2	Código Python y MQTT .....	29
3.9	Implementación de hardware .....	30
3.9.1	Implementación del teclado braille .....	30
3.9.2	Placa de protección .....	32
3.9.3	Diseño de la Estructura .....	32
3.9.4	Diseño electrónico.....	33
<b>CAPÍTULO 4.....</b>		<b>34</b>
<b>ANÁLISIS DE FUNCIONAMIENTO Y RESULTADO .....</b>		<b>34</b>
4.1.	Pruebas de idioma .....	34
4.2.	Pruebas de comunicación skill de Alexa a Lambda AWS .....	36
4.3.	Comunicación a la Raspberry Pi .....	37
4.4.	Pruebas con el usuario .....	38
4.5.	Validación del prototipo .....	46
4.6.	Costos de implementación del dispositivo. ....	47
4.7.	Comparación de precios con dispositivos comerciales .....	48
<b>CAPÍTULO 5.....</b>		<b>49</b>
<b>CONCLUSIONES.....</b>		<b>49</b>
<b>RECOMENDACIONES.....</b>		<b>51</b>
<b>REFERENCIAS .....</b>		<b>52</b>



## ÍNDICE DE FIGURAS

Figura 2.1. Estadísticas de discapacidad en el Ecuador .....	3
Figura 2.2. Teclado braille Portset braillekey G2 .....	6
Figura 2.3. Alfabeto en código braille .....	6
Figura 2.4. Signo de mayúscula .....	7
Figura 2.5. Signo de número .....	7
Figura 2.6. Signos de vocales acentuadas .....	8
Figura 2.7. Parámetros dimensionales del sistema braille .....	9
Figura 2.8. Solenoide push pull .....	13
Figura 2.9. Modulo regulador Step Down Lm2596 .....	14
Figura 3.1. Diagramas de bloques general del sistema .....	15
Figura 3.2. Diagrama de flujo general .....	16
Figura 3.3. Scripts de configuración de AVS Device SDK .....	18
Figura 3.4. Servicio IoT core .....	18
Figura 3.5. Interfaz para crear un objeto IoT .....	19
Figura 3.6. Claves y certificado proporcionada por AWS IoT .....	19
Figura 3.7. Direcciones para interactuar con el objeto IoT .....	20
Figura 3.8. Creación de una nueva Skill .....	21
Figura 3.9. Ventana principal para construir una habilidad .....	22
Figura 3.10. Slot Types .....	22
Figura 3.11. Endpoint .....	23
Figura 3.12. Funciones de AWS Lambda .....	24
Figura 3.13. Desencadenador Alexa Skill Kit .....	24
Figura 3.14. Archivo y Código index.js de la función Lambda .....	25
Figura 3.15. Función onIntent .....	26
Figura 3.16. Librería os.system .....	29
Figura 3.17. Sintetizador de voz gTTS .....	29
Figura 3.18. Configuración para comunicación a AWS IoT .....	29
Figura 3.19. Teclado matricial 8x5 .....	31
Figura 3.20. Diseño de las teclas en código braille .....	32
Figura 3.21. Placa de protección .....	32
Figura 3.22. Parte interna y externa del prototipo .....	33
Figura 3.23. Circuito electrónico .....	33
Figura 4.1. Pruebas con la usuaria Anabel con discapacidad visual .....	34

Figura 4.2. Prueba en la sección Test de la Alexa Developer Console.....	36
Figura 4.3. Estado de prueba entre la habilidad de Alexa y Lamba.....	36
Figura 4.4. Monitorización por medio de la plataforma AWS IoT.....	37
Figura 4.5. Mensajes recibidos por la Raspberry Pi .....	38
Figura 4.6. Mensajes recibidos de la evaluación de los números .....	38
Figura 4.7. Estadísticas de Discapacidad visual en la Provincia de Pichincha .....	39
Figura 4.8. Porcentaje de veces que se mencionó un número y porcentaje de fiabilidad de la skill .....	41
Figura 4.9. Sinónimos para el llamado de la letra C .....	41
Figura 4.10. Porcentajes de respuesta de la skill aprender abecedario .....	43
Figura 4.11. Datos estadísticos y comportamiento de la skill evaluación números...	44
Figura 4.12. Datos estadísticos y diagrama de la Skill evaluación abecedario.....	45

## INDICE DE TABLAS

Tabla 2.1. Características técnicas de placas de desarrollo.....	13
Tabla 3.1. Elementos utilizados y funcionalidad del prototipo.....	16
Tabla 3.2. Funciones de MQTT. ....	27
Tabla 4.1. Pruebas con el idioma español de México, España y Estados Unidos. ....	35
Tabla 4.2. Muestras tomadas de las principales habilidades de Alexa .....	40
Tabla 4.3. Muestras tomadas en la Skill aprender números.....	40
Tabla 4.4. Toma de muestras de la skill aprender abecedario .....	42
Tabla 4.5. Toma de muestras de la skill evaluación números.....	43
Tabla 4.6. Toma de muestras de la skill evaluación abecedario .....	45
Tabla 4.7. Porcentaje de validación del proyecto .....	47
Tabla 4.8. Tabla de costos.....	47
Tabla 4.9. Comparación de costos con dispositivos similares .....	48

## **RESUMEN**

El prototipo es una ayuda al aprendizaje de personas con discapacidad visual como para quienes deseen aprender el lenguaje braille, donde aprenderán mediante el desarrollo de un teclado auditivo y por medio de matrices conformadas por solenoides que también sirve para la retroalimentación, el prototipo será asistido por voz, controlado por una tarjeta electrónica llamada Raspberry Pi, así la interpretación será de manera interactiva, sencilla y de bajo costo que servirá para iniciar con el conocimiento del código braille. El dispositivo cuenta con un sistema electrónico dividido en dos zonas de aprendizaje y evaluación. La zona de lectura se acciona mediante un teclado matricial con cada tecla con su respectivo código en braille, donde se muestra el alfabeto distinguiendo las letras mayúsculas y números, con una tecla adicional para el signo generador. El usuario, presiona las distintas teclas con la respectiva simbología e inmediatamente se escucha el sonido con el nombre de la tecla correspondiente.

Para la segunda parte, se elaboró dos matrices formadas por diez solenoides que se despliegan cuando el usuario mediante comandos de voz las active así es posible la verificación de la escritura braille y evaluación de los caracteres.

Mediante pruebas de funcionamiento del prototipo, realizadas con personas con discapacidad visual, se obtuvo un 92% de efectividad y por medio de encuestas realizadas se obtuvo un porcentaje de validación del 90.17% en el dispositivo para la enseñanza del Sistema braille teniendo la finalidad de aprender y evaluar de una manera más dinámica con comandos de voz.

## **ABSTRACT**

The prototype is a learning aid for people with visual disabilities as well as for those who wish to learn the braille language, where they will learn through the development of an auditory keyboard and through two matrices made up of solenoids that also serve for feedback, the prototype will be assisted by voice, controlled by an electronic card called Raspberry Pi, thus the interpretation will be in an interactive, simple and low-cost way that will serve to start with the knowledge of the braille code. The device has an electronic system divided into two areas of learning and evaluation. The reading area is operated by means of a matrix keyboard with each key with its respective braille code, where the alphabet is shown distinguishing capital letters and numbers with an additional key for the generator sign. The user, presses the different keys with the respective symbology and immediately the sound with the name of the corresponding key is heard.

For the second part, two matrices formed by ten solenoids were elaborated that are displayed when the user activates them by means of voice commands, it is possible to verify the braille writing and evaluate each of the characters.

Through prototype functioning tests, carried out with people with visual disabilities, a 92% effectiveness was obtained and through surveys carried out, a validation percentage of 90.17% was obtained in the device for teaching the Braille system, with the purpose of learning and evaluate in a more dynamic way with voice commands.

## INTRODUCCIÓN

El presente proyecto, se desarrolló con la finalidad de iniciar el aprendizaje del lenguaje braille al ser un medio de comunicación tanto para la lectura y escritura, ya que las discapacidades que más afectación social tienen son la ceguera y la debilidad visual, razón por la cual la educación y desarrollo integral de estas personas se encuentran en desventaja es por ello que se crea un dispositivo embebido diseñado para enseñar los primeros pasos básicos del lenguaje.

El prototipo al ser una propuesta tecnológica, basa su implementación en una Raspberry Pi 3 B, el cual cuenta con un sistema operativo propio llamado Raspbian, también permite la conexión de parlantes para la salida de audio. La particularidad principal del sistema se realiza por medio de instrucciones de voz mediante el servicio de Amazon Alexa para un aprendizaje interactivo, facilitando el proceso de enseñanza.

El presente proyecto contiene la siguiente estructura de trabajo:

En el capítulo 1 presenta el planteamiento del problema, la justificación del proyecto, el objetivo general y objetivos específicos.

En el capítulo 2 se explican los conceptos principales del lenguaje braille, así como las herramientas para el desarrollo del dispositivo de manera general.

En el capítulo 3 se presenta el diseño y desarrollo del dispositivo, los elementos electrónicos que interviene para su implementación y herramientas para el hardware como software.

En el capítulo 4 se da a conocer los resultados obtenidos tanto pruebas de funcionamiento como la validación por parte de las personas con discapacidad visual.

En el capítulo 5, se muestran las conclusiones y recomendaciones del proyecto una vez alcanzado los objetivos.

# **CAPÍTULO 1**

## **ANTECEDENTES**

### **1.1. Planteamiento del Problema**

Para que una persona con discapacidad visual tenga acceso a una educación con un alto nivel de aprendizaje tiene un sin número de dificultades, una de ellas son los elevados costos de los productos que llegan al mercado, ya sea porque proceden de países extranjeros o debido a que el método de fabricación es considerado como artesanal y no de producción en serie. (Riazi, 2016)

El proceso inicial de escritura se realiza con una tabla manual de puntos en relieve, haciéndose necesario la supervisión de un especialista, lo cual limita a la persona al no ser un aprendizaje autodidacta y autónomo. (Hernández, 2016)

Entre los obstáculos a la educación para niños y adultos con discapacidad visual se localiza la falta de metodologías y materiales diseñados especialmente para las personas no videntes. Algunas estadísticas dicen que sólo el 10% de las personas consideradas ciegas o deficientes visuales utilizan el braille como sistema de enseñanza en lectura y escritura. El lenguaje braille permite involucrar a las personas no videntes tanto en el ámbito social, como laboral, debido a que es el medio de comunicación que ayuda a desempeñarse dentro de la sociedad (Castilla, 2015), es por ello que el presente proyecto plantea una solución tecnológica nacional a estas dificultades.

### **1.2. Justificación**

Actualmente para personas con discapacidad visual tengan acceso a la educación ha estado limitado por cuanto no existen centros de enseñanza públicas que presten este servicio, para lo cual sus familias han tenido que buscar a personas que tengan conocimientos de braille, y enseñen a niños, niñas y adolescentes teniendo un costo elevado. (Castilla, 2015)

Los dispositivos electrónicos dirigidos a personas que presentan discapacidad visual son de gran importancia, debido a que ayudan en su comunicación, educación

autónoma e independencia, además de la inclusión social. (Sánchez, 2016)

El impacto social de este dispositivo electrónico es más significativo, debido a que las personas con discapacidad visual estaban acostumbradas a depender de alguien más, hoy en día muchas personas invidentes que han logrado llevar una vida autónoma (Rivera, 2016), ya que en el año 2017 se emitió en Ecuador un decreto Ejecutivo 194 por el cual se expidió la ley orgánica de discapacidades en el Art. 12 habla sobre la inclusión laboral donde se exige que “el 2% de trabajadores con una discapacidad menor al 33%.” (Ley orgánica de discapacidades, N° 194, 2017)

En la actualidad en el mercado se han creado varios prototipos, pero a un muy elevado costo o con un sistema que no permite un aprendizaje autónomo, como sistemas para aprendizaje de lectoescritura en braille ABSé que tiene un costo de producción internacional de \$10.172 (Castañeda, 2015, pág. 27). El prototipo implementado tendrá la facilidad de que la persona podrá educarse y al mismo tiempo evaluar lo anteriormente aprendido sin necesidad de invertir gran cantidad de dinero.

### **1.3. Objetivos**

#### **1.3.1. Objetivo General**

Desarrollar un prototipo para aprendizaje de lenguaje braille grado 1, mediante un teclado asistido por voz y retroalimentación, permitiendo un aprendizaje interactivo.

#### **1.3.2. Objetivos Específicos**

- Investigar los sistemas de aprendizaje de lenguaje braille mediante fuentes de información como papers, libros y páginas web para la obtención de características del prototipo.
- Buscar información de sistemas de reconocimiento de voz que permita el menor porcentaje de error entre las distintas frecuencias de voz ingresadas.
- Diseñar la etapa de aprendizaje mediante el uso del teclado braille asistido por voz para un aprendizaje autónomo.
- Realizar un sistema de retroalimentación para la evaluación del aprendizaje.
- Implementar un prototipo que permita un aprendizaje interactivo y evaluación mediante comandos de voz.
- Realizar pruebas tanto de software como hardware para la validación de su correcto funcionamiento.



## CAPÍTULO 2

### FUNDAMENTACIÓN TEÓRICA

#### 2.1. Discapacidad visual

Según Tituana (2016) la discapacidad visual se considera a la pérdida ya sea parcial o total de la visión, que limita al individuo a la realización de sus actividades cotidianas.

Según la Organización Mundial de la Salud (OMS) existen cuatro categorías:

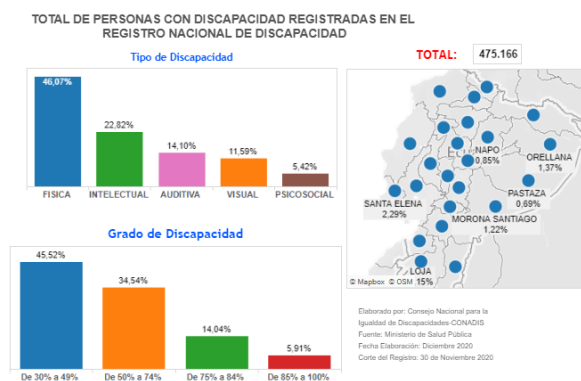
- Visión normal.
- Discapacidad visual moderada.
- Discapacidad visual grave.
- Ceguera.

La baja visión se reagrupa a la capacidad visual moderada y capacidad visual grave que son aquellas personas que tienen una disminución significativa pueden ver algo luz casi difusa y con sombras por otro lado la ceguera representa aquellas personas en ausencia total o mínima de la luz por lo que necesitan de la ayuda auditiva y táctil. (TITUANA, 2016, pág. 20)

#### 2.2. Discapacidad visual en el Ecuador

Según estadísticas obtenidas en diciembre 2020 por el CONADIS, como se muestra en la Figura 2.1 “En el Ecuador existen 475.166 personas registradas con algún tipo de discapacidad y de ellas 55.059 son no videntes”. (CONADIS, 2020)

Figura 2.1. Estadísticas de discapacidad en el Ecuador



Total, de personas con discapacidad registradas en el registro nacional de discapacidad. Fuente: (CONADIS, 2020)

Así sé que crea la Secretaría Técnica de Discapacidades o SETEDIS, para incluir a las personas con diferentes tipos de discapacidades como uno de los grupos primordiales que se debe tener en cuenta, siendo un acontecimiento de mayor relevancia para el Ecuador, ya que este grupo de personas fue apartado durante muchos años siendo ayudados por personas del sector privado o fundaciones. (Vicepresidencia de la República del Ecuador , 2019)

Según (GAIBOR, 2016), al presente en el Ecuador el cuidado a las personas con discapacidad se ha convertido una prioridad que se ve reflejada en el Artículo 53 el que avala la rehabilitación integral y la reinserción social de las personas con discapacidades. Este es uno de los artículos más importantes para las personas con discapacidad visual, ya que menciona las normas necesarias para exigir el cumplimiento de sus derechos.

### **2.3 Lenguaje braille**

“El método para la enseñanza del código braille se encuentra realizado mediante una celda conformada de 6 puntos de diferente tamaño, diversos materiales y realces, con lo cual el instructor intenta familiarizar al estudiante e iniciar la enseñanza de la escritura en este mismo sistema. Este sistema se utiliza desde inicios del siglo XIX con mínimos caracteres desde entonces.” (Haber-Olguin, 2015, págs. 35-36)

Según (Guzmán, 2016), la representación de los signos braille para las letras, números, signos de puntuación y también abreviaciones llamado estenografía. Se basa en la colocación de seis puntos, cuya forma y relieve son constantes y universales dentro de una celdilla o cajetín braille, creado por Louis Braille el cual número sistema braille en posiciones de los puntos en el cajetín: de arriba hacia abajo 1-2-3 al lado izquierdo y 4-5-6 al lado derecho formarán cada letra o signo específico, en cada celda se puede formar un solo carácter.

#### **2.3.1 Signo generador**

“Es una celda de puntos en relieve, basada en seis puntos de tres filas y dos columnas.” (MIRANDA, 2018).” Así mediante el cajetín es la base para generar las diferentes combinaciones de puntos que forman caracteres, y representar las letras del alfabeto, los números y los diferentes signos de puntuación. (Pauta, 2016)

## **2.4 Métodos de enseñanza y aprendizaje en niños**

Según Moya (2017), tanto los niños como adultos que empiezan su aprendizaje braille necesitan mayor atención de sus profesores especialmente más a ellos que presentan una deficiencia visual, hay que potenciar en el sentido del tacto, auditivos y cenestésicas. Las personas que presentan esta discapacidad, sus manos se tornan un órgano importante para su aprendizaje lo cual necesita entrenamiento para su desarrollo de este sentido.

Existen elementos de integración que ayudan al niño en su desarrollo de integridad y en la inclusión educativa, uno de estos es el educador de la clase, que necesita estar atento a todo lo que el realice niño para observar su avance. Una de las estrategias que se ha pensado es, animarlos para subir su autoestima y así se sientan como personas capaces de realizar actividades y labores como cualquier otra para al final sean una persona más de la sociedad. El educador debe ser seguro y esconder sus propias inseguridades para no causar ansiedad en el escolar una de la ayuda es trabajar con instrumentos determinados y didácticos, y ha de verbalizar todo lo que escriba. (Moya, 2017, pág. 24)

## **2.5 Teclado braille**

Mora (2016), describe a un teclado braille como un dispositivo que permite simbolizar cualquier signo mediante el pulso simultáneo de teclas, lo que permite lograr una mayor agilidad y facilidad en escritura. Los teclados braille tienen entre seis a ocho teclas, más una serie de teclas complementarias entre ellos existe los siguientes teclado braille.

- **Blue Type**

Teclado braille de 8 puntos donde tiene la característica Bluetooth y USB a distintos dispositivos electrónicos tanto sistemas operativos Windows como Linux.

- **Portset brailleKey**

Teclado braille de 6 puntos con conexión USB. Dispone además de las teclas para añadir una línea, dejar un espacio en blanco y retroceder.

## - Portset brailleKey G2

Como se muestra en la Figura 2.2, El teclado braille de 6 puntos con conexión USB que permite grado braille dos. Su característica es más avanzada ya que tiene teclas adicionales como, espacio en blanco, retroceso, Ctrl, Altr y los cuatro cursores. (Mora, 2016)

Figura 2.2. Teclado braille Portset braillekey G2

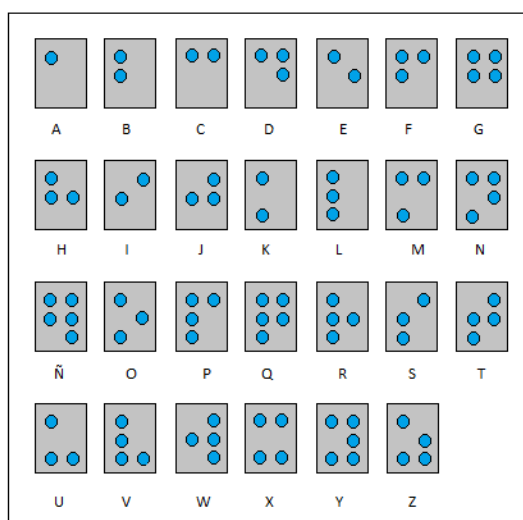


Teclado braille electronico. Fuente: Accesibilidad web.

## 2.6 Alfabeto braille

En la Figura 2.3 se muestran las 27 letras principales del alfabeto braille que permite leer a través del contacto, sobre una celda en relieve con puntos y guiones reconociendo las distintas letras. El alfabeto en código braille, consta de seis puntos conocidos como signo generador o universal, son de fácil aprendizaje y sus 64 combinaciones permiten representar el alfabeto, signos matemáticos, musicales, químicos, etc. (Escobar, 2017, pág. 91)

Figura 2.3. Alfabeto en código braille



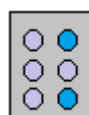
Alfabeto en código braille. Elaborado por: Paola Cabrera y Jennyfer Mejía.

## 2.7 Formaciones especiales del Sistema braille

### 2.7.1 Letras mayúsculas

“Para formar los caracteres en mayúsculas se coloca al comienzo de cualquier de letra minúscula del alfabeto, el símbolo formado por dos puntos en relieve, situados en la columna derecha como se observa en la Figura 2.4.” (COMISIÓN BRAILLE ESPAÑOLA, 2016, pág. 6).

Figura 2.4. Signo de mayúscula

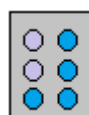


Signo generador para mayúsculas braille. Elaborado por: Paola Cabrera y Jennyfer Mejía.

### 2.7.2 Números

“En el sistema braille para los números se forma a partir de las diez primeras letras del abecedario de la “a” a la “j” con la diferencia que se coloca al inicio el símbolo de número en cada letra, a partir del 10 se realiza mediante la regla de alineación de la numeración, así para lograr una cifra de tres dígitos se realiza con los números del cero al nueve, pero el signo de número se escribe una vez al principio.” (MARTÍNEZ-LIÉBANA, 2015, pág. 43). La Figura 2.5 muestra el signo para números braille.

Figura 2.5. Signo de número

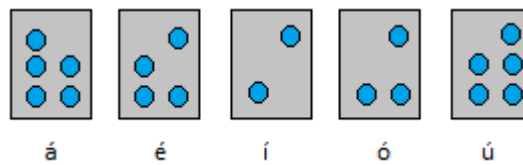


Signo generador para números. Elaborado por: Paola Cabrera y Jennyfer Mejía.

### 2.7.3 Vocales acentuadas

“Ya que no existe una simbología en braille para ubicar la tilde en las vocales se emplea una distribución de puntos distinto para las cinco vocales con tilde del abecedario español braille” se muestra en la Figura 2.6.: (MARTÍNEZ-LIÉBANA, 2015, pág. 45)

Figura 2.6. Signos de vocales acentuadas



Simbología para signos acentuados. Elaborado por: Paola Cabrera y Jennyfer Mejía

## 2.8 Grados del sistema braille

Según Sánchez (2016), en el sistema braille dependiendo del tipo de idioma y de acuerdo con los datos de la UNESCO, existe tres grados para el aprendizaje.

### 2.8.1 Grado Uno

Son aquellas palabras u oraciones que van de letra a letra para su escritura, lo cual inicia el aprendizaje de la persona no vidente. Al mismo tiempo este método sirve para formar con la educación del braille, porque permite una paciente comprensión de la escritura de cada símbolo por parte de la persona no vidente. (Sánchez, 2016, pág. 13)

### 2.8.2 Grado Dos

Radica en signos abreviados para comprimir el cuerpo de los escritos en braille y costos de impresión para preposiciones, pronombres, conjunciones, prefijos, sufijos. El grado dos intenta facilitar la escritura y lectura del sistema braille y facilita la comunicación de la persona no vidente. (FERRARO, 2015, pág. 25)

### 2.8.3 Grado Tres

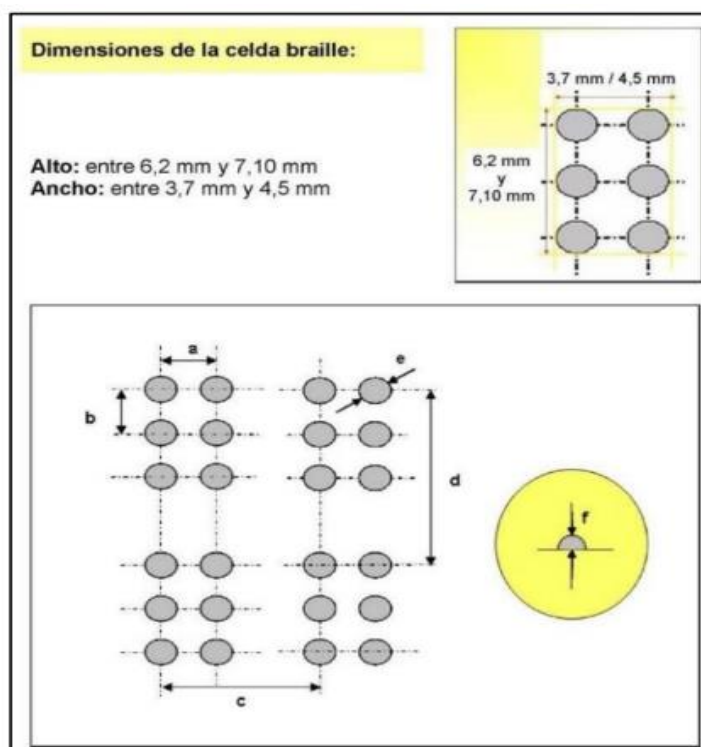
Constituye abreviaciones de palabras o pequeñas oraciones, lo cual implica un gran esfuerzo para escribir y comprender lectura de libros o revistas basados en este método, ya que necesita de una gran capacidad de memorización y sentido del tacto. (MIRANDA, 2018, pág. 8)

## 2.9 Parámetros dimensionales de los caracteres del Sistema braille

Para la dimensión de los caracteres son estandarizados por el Servicio Ecuatoriano de Normalización (INEN), norma NTE INEN2850 (INEN, 2014) y sus características se presenta en la Figura 2.7: (Escobar, 2017, pág. 92)

- $a$  = La distancia horizontal entre los centros de puntos adyacentes de la misma celda: de 2,4 a 2,75 mm.
- $b$  = La distancia vertical entre los centros de puntos adyacentes de la misma celda: de 2,4 a 2,75 mm.
- $c$  = La distancia entre los centros de puntos de las celdas contiguas: de 6,0 a 6,91 mm.
- $d$  = La distancia entre los centros de puntos iguales de líneas adyacentes: 10,0 a 11,26 mm.
- $e$  = El diámetro de los puntos: entre 1,2 y 1,9 mm.
- $f$  = La altura del relieve de los puntos: 0,50 mm.

Figura 2.7. Parámetros dimensionales del sistema braille



Parámetros dimensionales. Fuente: Revista Publicando, 4 No 12. (1). 2017, 89-107. ISSN 1390-9304.

## 2.10 Alexa voice service

Alexa voice service es el servicio de control de voz inteligente de Amazon Echo. Alexa utiliza técnicas de procesamiento del lenguaje realizado por los desarrolladores y la comunidad de Amazon para resolver las solicitudes de los usuarios y atender a sus necesidades basado en la nube de Amazon. El servicio de voz se puede activar usando la palabra clave Alexa. (Otero, 2018)

También distintas herramientas, APIs, como soluciones de referencia y documentación que contribuye de forma sencilla para que cualquier persona pueda crear con Alexa sin ningún costo. Además, tiene servicios de voz normalizados en el servidor y por lo tanto una unidad externa. (Kaundinya, 2017)

## **2.11 Componentes de AWS**

Según Tapashetti (2016), Amazon Web Services dispone de una amplia gama de productos en la nube como: análisis, IO, seguridad y almacenamiento.

### **2.11.1 AWS IoT**

AWS IoT es una plataforma en la nube que permite a los dispositivos conectados interactuar de manera fácil y segura con aplicaciones en la nube y hacia la plataforma cvAWS utilizando MQTT. AWS IoT puede tolerar miles de millones de dispositivos y mensajes, que los procesa y dirige a los puntos finales AWS y a otros dispositivos de forma fiable y segura. El usuario recibirá credenciales y una clave para acceder a los datos. El AWS ayuda a conectar el tablero a través de Wi-Fi a la cuenta específica proporcionando certificados, llaves públicas y privadas. (Tapashetti, 2016)

### **2.11.2 AWS Lambda**

Según Ms.Rajyashree (2018), Amazon Alexa depende de la nube pública de Amazon Web Services (AWS) donde un desarrollador puede subir el código en base a las capacidades de Alexa y a las funciones AWS Lambda para ejecutar y modificar el código que se activa por interacciones de voz. AWS gestiona automáticamente los recursos de cifrado para Lambda. Un desarrollador certificará, publicará y actualizará las habilidades, que se crean allí a través de la Alexa Skills Store que se activará mediante el ID de la aplicación así Lambda activa el dispositivo IoT (Internet of Things) escuchando los comandos de voz del usuario. (p.828)

Lambda tiene lenguaje de propagación como Node.js, Python, Java, Go, Ruby y C legalmente a partir de 2018. A finales de 2018, se agregó a AWS Lambda, dando a los desarrolladores la capacidad de ejecutar un Lambda en el idioma que elijan. AWS Lambda admite la ejecución segura mediante llamadas desde lenguaje compatible como Node.js. (Ms.R.Rajyashree, 2018, pág. 831)



### **2.11.3 AWS EC2**

Elastic Compute Cloud (EC2) es un servicio web sencillo que ayuda a elaborar máquinas virtuales en la nube. Es fiable y controlable. Como su nombre lo dice, permite a las empresas ejecutar programas de aplicación en la nube pública de Amazon. (Anoja Rajalakshmi, Hamid Shahnasser, 2017)

Un usuario de AWS puede incrementar o disminuir la capacidad de la petición según sea necesario en minutos utilizando la interfaz web de Amazon EC2 o una interfaz de programación de aplicaciones (API). (Rouse, 2019)

### **2.11.4 Alexa Skill**

El Skill de Alexa es un kit de desarrollo de sistemas de software (SDK) que permite a un desarrollador crear habilidades, conocidas conjuntamente como aplicaciones coloquiales, en el asistente de inteligencia artificial Alexa del Amazon. El Alexa Skills está compuesto por herramientas, interfaces de programa (API), muestras de código y documentación que permite a un desarrollador añadir habilidades a las más de 10.000 capacidades de reconocimiento de voz disponibles en Alexa. (Ms.R.Rajyashree, 2018)

## **2.12 MQTT**

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería simple diseñado especialmente para dispositivos. El protocolo trabaja como publicador-suscriptor en el que un bróker funciona como medio para distribuir los mensajes 'de uno a muchos'. La función primordial es disminuir los requerimientos que usan el ancho de banda de red y los recursos de los dispositivos, conservando cierto grado de fiabilidad, lo que hace que su uso en IoT o conexiones machine-to-machine sea beneficioso. (Castillo, 2019)

### **2.12.1 Características de MQTT**

El protocolo tiene diferentes características las cuales son:

- Es un protocolo de publicación/suscripción.
- No depende del contenido del mensaje.
- Se establece sobre TCP/IP, que suministra una conexión de red básica.
- En el caso de un fallo de conexión de un cliente de un servidor MQTT se notifica a los suscriptores.
- En MQTT cada conexión se conserva abierta y se «reutiliza» en cada comunicación.

### **2.13 JSON**

Alexa Skill trabaja con el lenguaje de programación JSON que es una plataforma que ofrece Amazon. JSON “Java Script Object Notation” permite acumular la información de una manera fácil y organizada basado en texto plano para el cambio de información, que requieren mostrar o enviar información para ser interpretada por otros.

JSON permite transmitir datos que consisten en atributos y valores. Es el formato de datos principal utilizado para la comunicación asíncrona del navegador-servidor, reemplazando en gran medida a XML. La mayor densidad de datos de JSON es la mejor opción para las aplicaciones IOT (Diamantaras, 2019)

### **2.14 Node.js**

Node.js es un programa para el desarrollo para programar tanto el cliente como el servidor, mediante la interfaz de Lambda para elaborar mediante JavaScript del lado del servidor de manera rápida y escalable. El objetivo principal de Node, es suministrar una manera fácil para desarrollar programas que contengan la propiedad de escalabilidad. (GONZAGA, 2015)

NodeJS al momento de ejecutar el código se realizan peticiones que requieren un tiempo para originar dicha petición, el cual no detiene la ejecución, sino que sigue procesando a las instrucciones y al terminar, realiza los comandos específicos para obtener los resultados aceptados. (Flore, 2017)

### **2.15 Raspberry Pi modelo B +**

El prototipo cuenta con una tarjeta electrónica llamada Raspberry Pi 3, debido a sus características y velocidad de procesamiento fue el indicado para usarlo en el proyecto porque el almacenamiento requiere de una tarjeta microSD, sólo debe conectarse a la corriente eléctrica, cuenta con hasta 4 puertos USB para conectar teclado y mouse, un conector HDMI con capacidad de reproducir vídeo en 1080p, WiFi, entradas y salidas GPIO (Entrada/Salida de propósito general) y conexión Ethernet para poder tener internet vía cable. También es un dispositivo ideal para el desarrollo de proyectos IoT. (Pastor, 2018)

En la Tabla 2.1 se observa las características de distintos controladores embebidos donde las principales diferencias se aprecian en el procesador.

Tabla 2.1. Características técnicas de placas de desarrollo.

ELEMENTO	ARDUINO UNO	Raspberry Pi 1-B	Raspberry Pi 3-B
Velocidad de reloj	16 MHz	700 MHz	1,2 GHz
# núcleos	1	1	4
RAM	0,002 MB	512MB	1GB
CPU	no	ARM1176JZF-S	ARM Cortex-A53
SOC	ATmega328P	Broadcom BCM2835	Broadcom BCM 2837
Flash	32KB	Tarjeta SD (2-16 Gb)	Tarjeta microSD (2-32 Gb)
SO	Ninguno	Distribuciones Linux	Distribuciones Linux
Cámara	NO	No	5Mp
Precio	22,88	34,19	51,29
Tamaño	7,6X1,9X6,4cm	8,6X5,4X1,7cm	8,6X5,4X1,7cm
Peso	40 gr	45 gr	45gr
Pines	14	40	40

Detalle de las características de los sistemas embebidos para seleccionar el ideal para el sistema de visión artificial. Fuente: **Fuente especificada no válida.**

## 2.16 Solenoides

Los solenoides son electroimanes, fabricados de una bobina de alambre de cobre con una armadura de metal en el medio, tal como se muestra en la Figura 2.8. Funciona cuando se energiza a 12 voltios creando un campo magnético y esto hace que el pistón del solenoide sea capaz de salir. Están colocados de forma que se cree el signo generador y una cuadrilla para cada carácter, en el cual la persona no vidente a través de sus manos podrá detectar qué posición está activa y de esta manera ir reconociendo consonante sean estas mayúsculas o minúsculas y números.

Figura 2.8. Solenoide push pull



Solenoides de Corriente Continua. Elaborado por: Paola Cabrera y Jennyfer Mejía

## 2.17 Módulo conversor DC/DC

Este módulo regulador está apoyado en el chip LM2596, como se observa en la Figura 2.9 que es un circuito integrado monolítico, útil para una fuente de tipo Buck. Con una corriente de hasta 3A y de regular el voltaje de entrada de un circuito a partir

de un voltaje alto. El módulo disminuye el uso de elementos extras para el diseño de fuentes de alimentación. (HIDALGO, 2018)

Figura 2.9. Módulo regulador Step Down Lm2596



Módulo DC/DC Lm2596 3.7V-5V. Fuente: (HIDALGO, 2018)

## CAPÍTULO 3

### DISEÑO E IMPLEMETACIÓN

El presente capítulo trata sobre la implementación del prototipo a partir de un mecanismo basado en un teclado matricial para el sistema audible en código braille y diez solenoides que serán controladas por medio de comandos de voz mediante el servidor de Amazon Alexa, todo ello es automatizado por medio una Raspberry Pi que funciona como dispositivo IoT AWS y Lambda, para la instrucción a la enseñanza del código en braille dirigido a niños, jóvenes y adultos, no videntes. Asimismo, el sistema también puede ser utilizado para enseñar a personas videntes.

#### 3.1 Diagrama de bloques

El diagrama de bloques que se muestra a continuación indica el camino para llegar a controlar la Raspberry Pi desde Alexa Voice Service el cual para ejecutar el sistema se necesita de Alexa Voice y Paho MQTT controlados al mismo tiempo ya que se puede utilizar la Raspberry Pi para hacer la funcion de Amazon ECHO para Alexa Voice Service y cliente MQTT. La Figura 3.1 muestra cómo funciona el sistema.

Figura 3.1. Diagramas de bloques general del sistema.

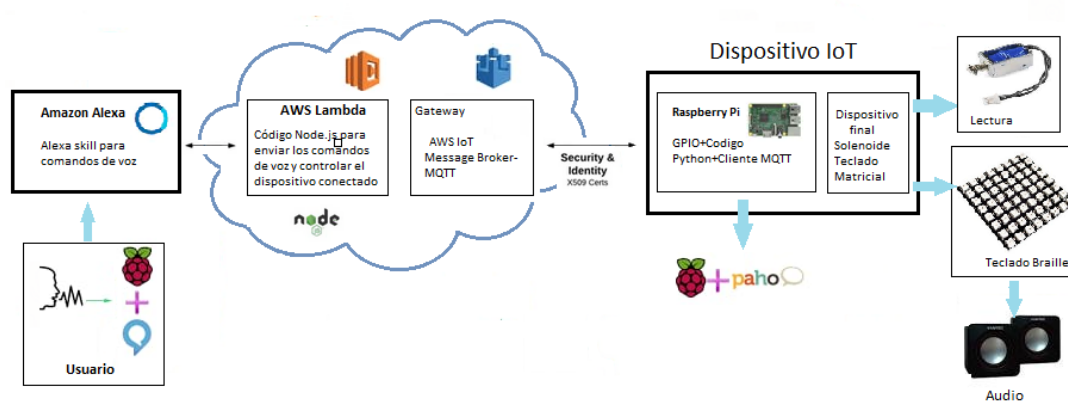


Diagrama general del sistema. Elaborado por: Paola Cabrera y Jennyfer Mejía.

En la Figura 3.1 muestra al usuario emitiendo un comando de voz para realizar una tarea específica este a su vez va a ser interpretada por la habilidad de Alexa donde se encuentra las palabras claves para llamar a la skill de Alexa y controlar los dispositivos. A continuación, se requiere de la plataforma AWS Lambda donde se

almacena y ejecuta el código desde el servidor de Amazon para lo cual se necesita cargar el código fuente y sus dependencias de Node.js.

La función *Lambda* se ejecuta y envía un mensaje específico a la nube de IoT mientras la tarjeta electrónica Raspberry Pi va a recibir un mensaje que contenga información para el control del dispositivo de acuerdo con su comando de voz definido en la skill.

Para controlar e interactuar con el dispositivo desde cualquier ubicación se necesita el servicio en la nube IoT que soporte MQTT con SSL que requiere un cliente MQTT creado en la Raspberry Pi la cual comunica al servidor por medio AWS IoT además de proporcionar certificados y credenciales que se colocaran tanto el código de Python como *Lambda* para proteger la comunicación. Así una vez que Alexa se activa, ejecuta un script en la nube, y al mismo tiempo una subrutina en la Raspberry Pi. De esta manera el mensaje enviado por la confirmación a Alexa lo transmite para activar los dispositivos finales desde las salidas GPIO de la Raspberry Pi.

A continuación, en la Tabla 3.1 se detalla los elementos utilizados para ser implementados en el sistema, así también el software y su funcionalidad.

Tabla 3.1. Elementos utilizados y funcionalidad del prototipo.

<b>Materiales</b>	<b>Función</b>
Raspberry Pi 3 B	Se encarga de ser un dispositivo IoT para enviar y recibir datos MQTT además de activar y desactivar los actuadores
Código Python	Se programa las entradas y salidas de la Raspberry Pi así como el servidor MQTT.
Solenoides push pull	Reciben las señales para ser activadas según el código braille.
Teclado matricial	Recibe las señales para activar las teclas según el signo braille y emitir su sonido.
AWS IoT	Crea los certificados X.509 y Message Broker con el uso del protocolo MQTT
Alexa Skill	Alexa se activa, ejecuta un script en la nube
AWS Lambda	El código fuente se activa por medio Alexa Voice Service para transmitir las instrucciones de Alexa.
Node.js code	Código permite Node.js que actúe como un servidor web sin software adicional.
Amazon Echo	Usa el servicio de voz de Alexa (AVS) para agregar control de voz inteligente a cualquier producto conectado que tenga un micrófono y un altavoz.

Detalle de elementos y funciones del dispositivo. Elaborado por: Paola Cabrera y Jennyfer Mejía

### 3.2 Funcionamiento lógico del dispositivo

En la Figura 3.2 se observa el diagrama de flujo general del dispositivo, basado en dos etapas de aprendizaje y evaluación del Código braille, así el usuario podrá ingresar e interactuar con la aplicación del sistema.

Figura 3.2. Diagrama de flujo general

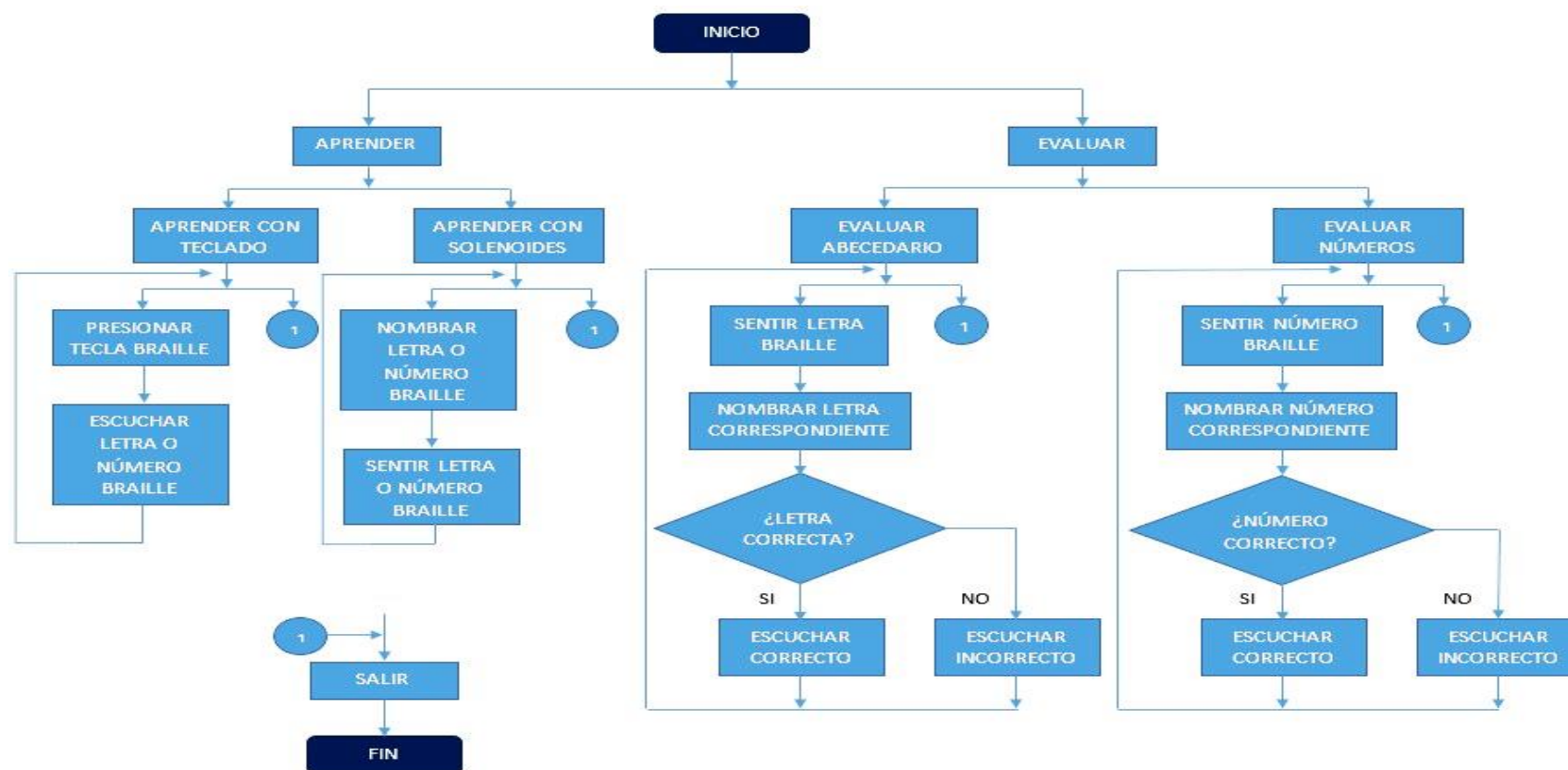


Diagrama general del prototipo. Elaborado por: Paola Cabrera y Jennyfer Mejía.

### 3.3 Instalación y preparación de Raspberry Pi

Antes crear un dispositivo IoT en la Raspberry Pi 3 B para el presente proyecto se necesita la instalación del sistema operativo “Raspberry Pi OS” compatible y oficial del dispositivo, para descargar la imagen con una gran cantidad de software para programación y uso general como Python, Scratch, Sonic Pi, Java y más.

Una vez que el dispositivo se encuentra en funcionamiento, se procede a configurarlo utilizando el código de Python, por lo que es muy importante verificar e instalar otros paquetes y bibliotecas necesarios para la Raspberry Pi. Python 2.7 está instalado de manera predeterminada y se puede ser verificado con el comando en la terminal “*Python - -version*” o también revisando la lista completa de los paquetes instalados por defecto usando “*apt-mark showauto*”

Además de Python, se necesita 4 paquetes para el código, que son, “*time*”, “*openssl*”, “*rpi.gpio*” y “*paho.mqtt.client*”. De las cuales, el paquete “*paho.mqtt.client*” no está disponible por defecto. Se Puede instalar en la Raspberry Pi se utiliza la siguiente línea de comando: “*pip install paho-mqtt*”.

### 3.4 Instalación del Servidor de voz Alexa en la Raspberry Pi

El servidor de voz de Alexa proporciona un software de voz de Amazon compatible con dispositivos inteligentes como la Raspberry que permite automatizar y ser usado como un Amazon echo. Antes de instalar se crea un producto AVS y un perfil de seguridad, al final se descarga un archivo *config.json* que contiene el ID de cliente y *client secret* que autorizan sus interacciones con Alexa y que sirven para el acceso a los tokens y actualización.

A continuación, en el terminar de la Raspberry se copia los scripts desde el repositorio GitHub para la configuración de AVS Device SDK como se muestra en la Figura 3.3 que contienen toda la lógica para construir el SDK así al ejecutarse se descarga e instalan las dependencias necesarias también automáticamente de instala la última versión SDK.



Figura 3.3. Scripts de configuración de AVS Device SDK

```
wget https://raw.githubusercontent.com/alexa/avs-device-  
sdk/master/tools/Install/setup.sh \  
wget https://raw.githubusercontent.com/alexa/avs-device-  
sdk/master/tools/Install/genConfig.sh \  
wget https://raw.githubusercontent.com/alexa/avs-device-  
sdk/master/tools/Install/pi.sh
```

Scripts de instalación y configuración de SDK. Elaborado por: Paola Cabrera y Jennyfer Mejía

Una vez descargado los archivos de configuración para la instalación en la Raspberry Pi se debe utilizar el archivo *setup.sh*, *config.json* que contiene las credenciales y un número de serie del dispositivo (DSN) como argumentos como se muestra con el siguiente comando: *sudo bash setup.sh config.json [-s 1234]*

Una vez finalizado se genera un script llamado *startsample.sh* para iniciar la aplicación y solicitar un token de autorización para el servidor de voz de Alexa se realiza mediante el comando *sudo bash startsample.sh*.

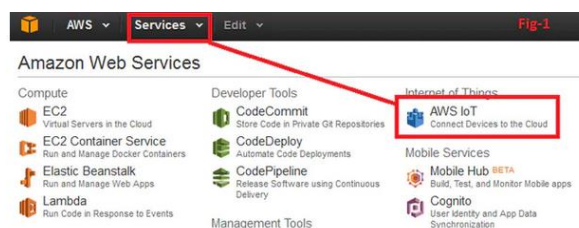
### 3.5 Configuración AWS IoT

Todos los servicios se entregan a través de la consola de administración de AWS basada en la web. Se debe tener presente los siguientes detalles creando su objeto:

- Punto final de AWS IoT personalizado: "xxxxxxxxxxxxxxxx-ats.IoT.eu-west-1.amazonaws.com"
- Región - ejemplo: "us-east-1"
- ID del dispositivo: "acuario"
- Certificados: "cert.pem, private.key y AmazonRootCA1.key"

Para comenzar se necesita iniciar la sesión en la cuenta de AWS de Amazon y desde la pestaña “Servicios”, se debe abrir **"IoT Core"**. En la Figura 3.4 se observa el proceso antes mencionado.

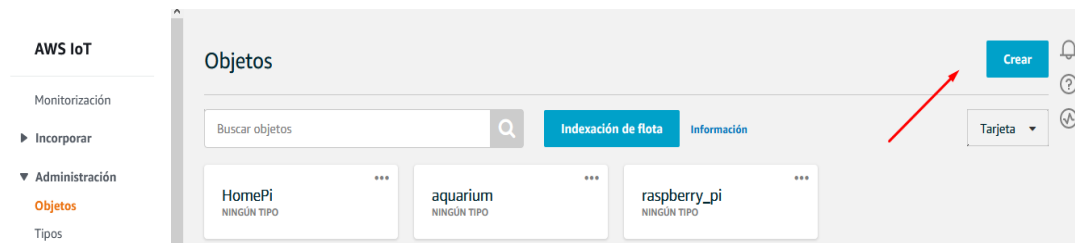
Figura 3.4. Servicio IoT core



IoT Core de AWS. Elaborado por: Paola Cabrera y Jennyfer Mejía

Como se observa en la Figura 3.5 dentro de la ventana "IoT Core", se debe ingresar a través de la pestaña "Administración" a "Objetos" y seguidamente clic en "Crear"

Figura 3.5. Interfaz para crear un objeto IoT



Creación de un objeto IoT. Elaborado por: Paola Cabrera y Jennyfer Mejía

Una vez creado, se define el nombre y registra el objeto. En la Figura 3.6 se observa la ventana para descargar el certificado y archivos que contienen las, proporcionada por AWS IoT. Posteriormente se activa el certificado y se crea una política. Estos certificados son necesarios tanto en la Raspberry Pi y con el código Lambda.

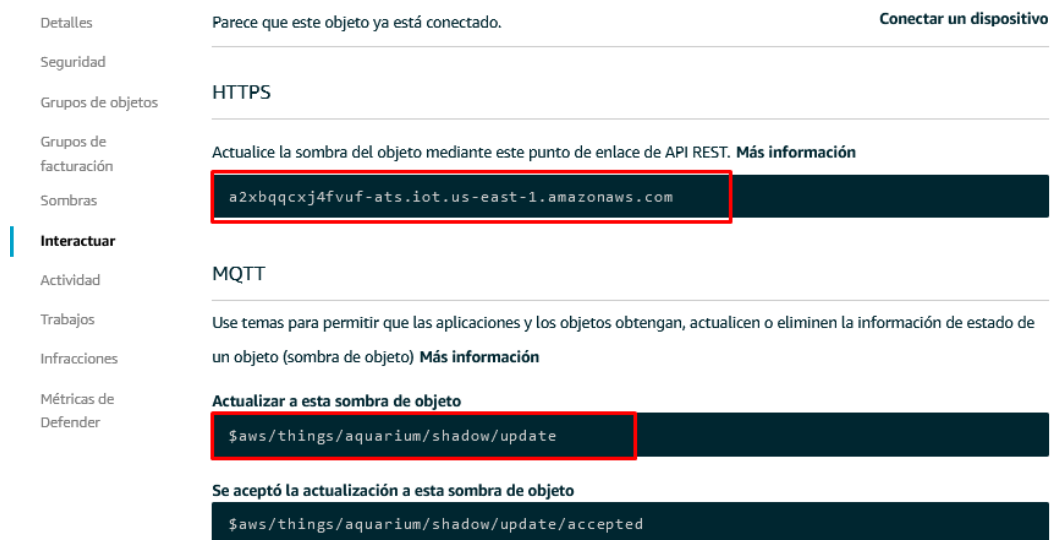
Figura 3.6. Claves y certificado proporcionada por AWS IoT



Certificados para la clave privada y la clave de CA raíz. Elaborado por: Paola Cabrera y Jennyfer Mejía

Para encontrar el objeto IoT personalizado o nombre de "host", se debe navegar hasta el icono "Interactuar". En la Figura 3.7 se aprecia la dirección HTTP y MQTT para el código en Python e "index.js" de Lambda.

Figura 3.7. Direcciones para interactuar con el objeto IoT



Detalle del objeto IoT personalizado o nombre de "host". Elaborado por: Paola Cabrera y Jennyfer Mejía

Por último, se debe regresar a la Raspberry Pi y crear la carpeta "certs" para copiar los tres certificados nombrados como "private.pem.key", "certificate.pem.crt" y "AmazonRootCA1.key". La ubicación de la carpeta antes mencionada y los nombres de los archivos, deben ser mencionados en el código.

El archivo de Python se proporciona los principales cambios en los que te centra el código los cuales son:

- "cert\_path" = "/home/pi/Desktop/certs3/"
- "host" = "a2xbqqcxj4fvuf-ats.iot.us-east-1.amazonaws.com"
- "tema" = "\$ AWS / things / abecedario / shadow / update"
- Nombres del certificado raíz, archivo cert y archivo clave

### 3.6 Creación de la habilidad personalizada de Alexa

Las interfaces de programa de aplicaciones (API) de la cuenta desarrolladores de Amazon se crean por medio de una habilidad de Alexa o Alexa Skill, donde se definen y procesan los comandos de voz antes de enviarlos a un servidor local alojado en la Raspberry Pi.

Para comenzar se crea una nueva cuenta en developer.amazon.com. Se debe tener en consideración en usar el mismo usuario y contraseña de Amazon de lo contrario no podrá interactuar con el dispositivo de Alexa.

Al iniciar la sesión se selecciona Alexa Skill kit. En la Figura 3.8 se muestra la ventana para crear una Skill donde se escribe el nombre y se escoge el idioma. Luego se va a elegir un modelo para agregar a la habilidad, existen plantillas ya preconstruidas por defecto, pero para realizar un diseño personalizado se debe elegir “custom”. Luego en “Create a Skill”.

Figura 3.8. Creación de una nueva Skill

alexa developer console

Create a new skill

Cancel Create skill

Model: Custom  
Host: Alexa-Hosted (Prode26)  
Hosting Region: US East (N. Virginia) v

Skill name  
Enter skill name 0/50 characters

Default language  
Spanish (ES) More languages can be added to your skill after creation

1. Choose a model to add to your skill  
There are many ways to start building a skill. You can design your own custom model or start with a pre-built model. Pre-built models are interaction models that contain a package of intents and utterances that you can add to your skill.

Custom  
Design a unique experience for your users. A custom model enables you to create all of your skill's interactions. **SELECTED**

Flash Briefing  
Give users control of their news feed. This pre-built model lets users control what updates they listen to.  
"Alexa, pon el resumen de noticias."

Smart Home  
Give users control of their smart home devices. This pre-built model lets users turn off the lights and other devices without getting up.  
"Alexa, enciende las luces de la cocina"

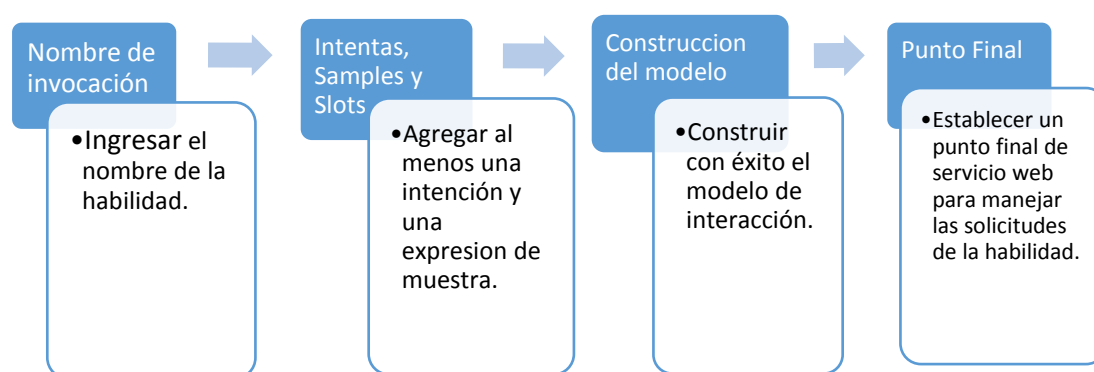
Video  
Let users find and consume video content. This pre-built model supports content searches and content suggestions.  
"Alexa, pon Interstellar"

Creación de una nueva Skill de Amazon. Elaborado por: Paola Cabrera y Jennyfer Mejía

En la Figura 3.9 se muestra los pasos a seguir para construir la habilidad, donde se inicia estableciendo el nombre de invocación y así llamar a la Skill con la frase “Alexa iniciar (nombre de invocación)”, los “Intents” donde contiene los enunciados de muestra y para un intento determinado “samples”.

El editor JSON (JavaScript Object Notation) que contiene la matriz de los objetos, para ver y editar todos sus intentos y expresiones, los “Slot Types” que son las diferentes expresiones o diferentes formas de referirse a un objeto que el usuario utiliza para interactuar como modelo de diálogo, por últimos se define el punto final o dirección para la comunicación con Alexa y Lambda y responder las solicitudes de la habilidad.

Figura 3.9. Ventana principal para construir una habilidad



Creación de una nueva Skill ventana principal de Amazon. Elaborado por: Paola Cabrera y Jennyfer Mejía

A continuación, se define las “*Intents*” o propósitos que representan una operación que efectúa la solicitud del usuario. Los intentos pueden tener argumentos llamados “Slot” los cuales se utilizan después de escribir algunas palabras o frases que simbolizan información. Los “*Sample Utterances*” son un conjunto de posibles frases habladas asignadas a los intentos que son ejemplos que se pide para realizar dichas acciones.

Ya creado los “*Intents*” se debe definir los “*Slot Types*” representados en la lista de posibles valores incluyendo el ID y sinónimos escritos opcionalmente tal como se muestra en la Figura 3.10, se tomó como ejemplo la Skill que contiene los números en base 10. Se debe escribir la palabra, sílaba o fonema para que la habilidad de Amazon Alexa tome estos valores mediante una entrada de voz los cuales va a traducir a texto por los servidores de Amazon y los compara, posteriormente envía la respuesta de JSON a una aplicación o servidor MQTT.

Figura 3.10. Slot Types

VALUE ⓘ	ID (OPTIONAL) ⓘ	SYNONYMS (OPTIONAL) ⓘ		
dos	Enter ID	Add synonym	+	2 x dos x
uno	Enter ID	Add synonym	+	uno x 1 x

Lista de valores definidos por *Slot Types*. Elaborado por: Paola Cabrera y Jennyfer Mejía

Como último paso se debe escribir la ruta por la cual se comunica con AWS Lambda por medio de “Endpoint” como se observa en la Figura 3.11 se ingresa el ARN que es recomendado al momento de crear una habilidad ya que no es necesario modificarlo. Se debe tener en cuenta el ID de habilidad porque es necesario cuando se cree la función Lambda.

Figura 3.11. Endpoint

Endpoint

The Endpoint will receive POST requests when a user interacts with your Alexa Skill. The request body contains parameters that your service can use to perform logic and generate a JSON-formatted response. Learn more about AWS Lambda endpoints here. You can host your own HTTPS web service endpoint as long as the service meets the requirements described here.

Service Endpoint Type

Select how you will host your skill's service endpoint. Best practices in choosing lambda regions. [Learn more here](#)

☒ AWS Lambda ARN (Recommended)

Your Skill ID ⓘ amzn1.ask.skill.dbb9f2e5-a833-48a3-b965-ed19de95a418  
Copy to Clipboard

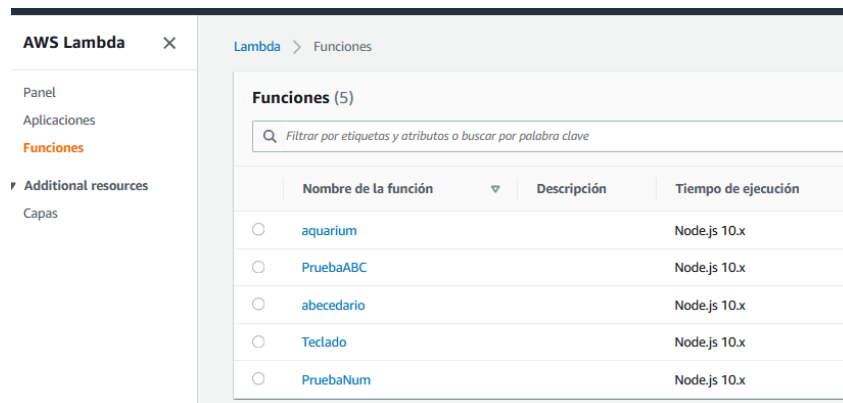
Default Region (Required) ⓘ arn:aws:lambda:us-east-1:397968631940:function:PruebaNum

Punto final e ID para realizar la conexión con AWS Lambda. Elaborado por: Paola Cabrera y Jennyfer Mejía

### 3.7 AWS Lambda

En el presente proyecto se desarrolló cinco funciones *Lambda* desde la consola de AWS como se observa en la Figura 3.12. Por lo que es necesario el nombre la función, el tiempo de ejecución donde se elige el lenguaje, el cual se escogió “Node.js 10x” y el rol de ejecución que define los permisos de la función, para la función se creó un nuevo rol con el nombre *Lambda\_basic\_execution*.

Figura 3.12. Funciones de AWS Lambda



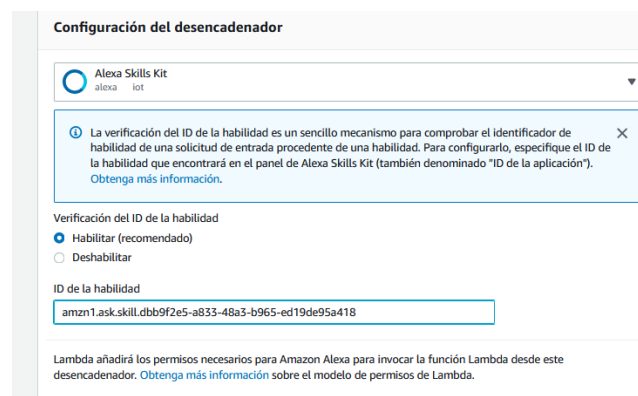
AWS Lambda			
Panel			
Aplicaciones			
Funciones			
Additional resources			
Capas			
Lambda > Funciones			
Funciones (5)			
Filtrar por etiquetas y atributos o buscar por palabra clave			
	Nombre de la función	Descripción	Tiempo de ejecución
<input type="radio"/>	aquarium		Node.js 10.x
<input type="radio"/>	PruebaABC		Node.js 10.x
<input type="radio"/>	abecedario		Node.js 10.x
<input type="radio"/>	Teclado		Node.js 10.x
<input type="radio"/>	PruebaNum		Node.js 10.x

Cinco funciones AWS Lambda. Elaborado por: Paola Cabrera y Jennyfer Mejía

Cabe resaltar que cada una de las funciones tiene una habilidad de Alexa propia. La primera función con el nombre de “*Aquarium*” contiene el código de los números en base 10, la segunda es “*Abecedario*” tiene la llamada de la función para cada una de las letras del abecedario, la tercera es “*Teclado*” como su nombre lo indica para activarlo o desactivarlo, por último, “*PruebaABC*” y “*PruebaNum*” donde se evaluará de los números y letras aprendidos en código braille. Todas estas funciones son llamadas cuando el usuario especifica una intención de cada habilidad.

Ya creada la función se debe añadir un desencadenador, AWS Lambda ofrece diferentes opciones. Para realizar la comunión con la habilidad de Alexa se escoge “*Alexa Skill Kit*” y se coloca la ID de la habilidad tal como indica en la Figura 3.13.

Figura 3.13. Desencadenador Alexa Skill Kit



Configuración del desencadenador

Alexa Skills Kit

La verificación del ID de la habilidad es un sencillo mecanismo para comprobar el identificador de habilidad de una solicitud de entrada procedente de una habilidad. Para configurarlo, especifique el ID de la habilidad que encontrará en el panel de Alexa Skills Kit (también denominado "ID de la aplicación"). [Obtenga más información.](#)

Verificación del ID de la habilidad

☒ Habilitar (recomendado)

☐ Deshabilitar

ID de la habilidad

amzn1.ask.skill.dbb9f2e5-a833-48a3-b965-ed19de95a418

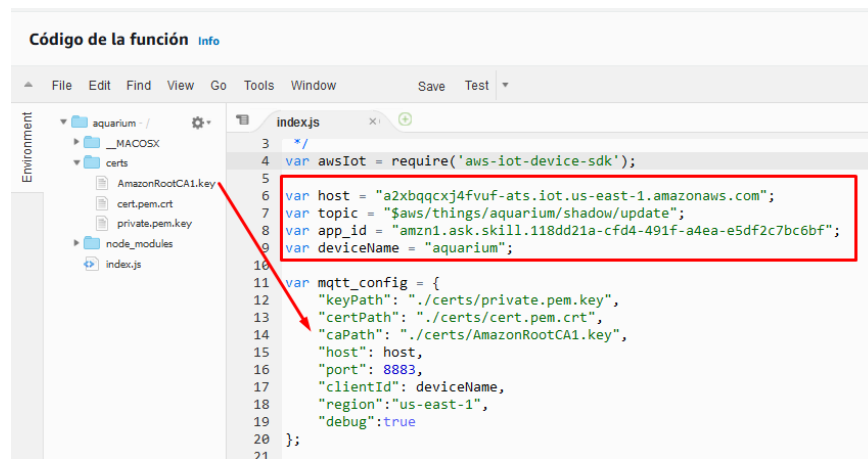
Lambda añadirá los permisos necesarios para Amazon Alexa para invocar la función Lambda desde este desencadenador. [Obtenga más información](#) sobre el modelo de permisos de Lambda.

Desencadenador Alexa Skill Kit. Elaborado por: Paola Cabrera y Jennyfer Mejía

AWS Lambda por defecto crea un archivo *index.js* para empezar a programar, pero antes se debe cargar un archivo ZIP que contiene el código y las dependencias de la función, necesarios para Node.js. Así AWS Lambda ejecuta aplicaciones, en el directorio *node\_modules* del lenguaje de programación. La implementación del código Node.js en AWS se obtuvo de (Developer, 2010-2021). También se debe cargar los certificados obtenidos anteriormente en AWS IoT Core.

En la Figura 3.14 se muestra los principales cambios que se enfoca “index.js” así como las carpetas con los certificados de AWS IoT y las dependencias de Node. Es muy importante el nombre de los certificados coinciden con el código al igual de la ubicación en donde estos se encuentren.

Figura 3.14. Archivo y Código index.js de la función Lambda



Certificados AWS IoT, dependencia de Node.js y archivo index. Elaborado por: Paola Cabrera y Jennyfer Mejía

Para realizar la comunicación entre Alexa y Lambda se necesita la ID de la Skill y para la conexión de AWS IoT se necesita el nombre de “Host”, “Topic” y nombre del dispositivo obtenidos en creación del objeto IoT como se indica en la Figura 3.14. También es necesario el nombre de los certificados, así como el puerto por defecto 8883 de MQTT y la región que se encuentra la consola de AWS.

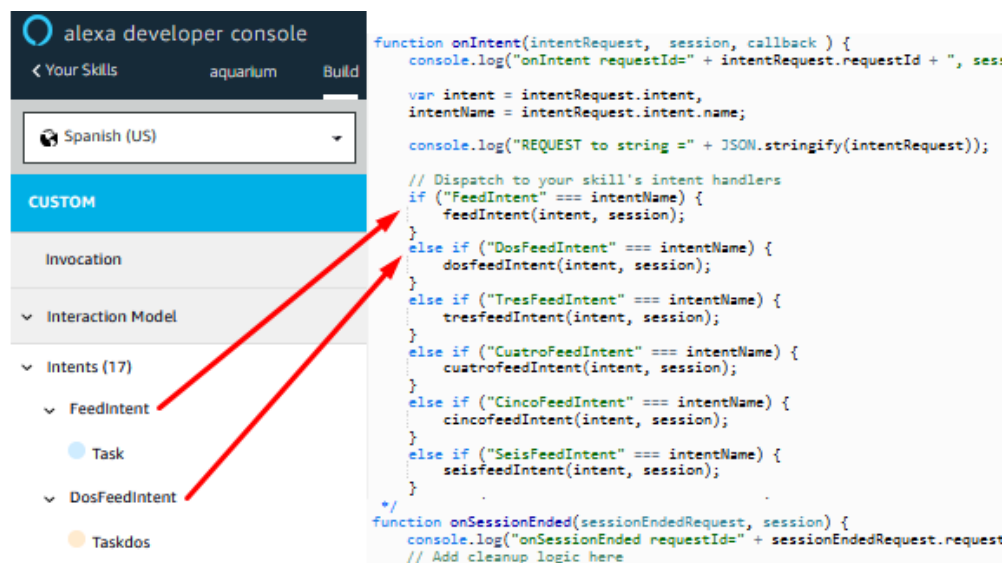


### 3.7.1 Etapa de aprendizaje

#### 3.7.1.1 Función AWS Lambda para las Solenoides

En la etapa de aprendizaje se creó las funciones con el nombre de “*Aquarium*” y “*Abecedario*”, se tomará como ejemplo la primera función mencionada anteriormente. La primera parte del código Node.js escrito *JavaScript* enrutará la solicitud entrante en función de su tipo. El tipo será “*LaunchRequest*”, “*IntentRequest*” o “*SessionEndedRequest*” y esta función se activará cada vez que interactúe con Alexa. Así también con el cliente de MQTT y conectara al dispositivo de AWS IoT. El cuerpo JSON de la solicitud se proporciona en el parámetro del evento. La función “*onIntent*” sirve cuando se declara una intención específica de la Skill de Alexa como se observa en la Figura 3.15, se debe tener en cuenta en escribir los mismos nombres de los “*Intents*” de la habilidad y “*onSessionEnded*” se llama, cuando el usuario finaliza la sesión.

Figura 3.15. Función onIntent



Función *onIntent* para llamar a un *Intent*. Elaborado por: Paola Cabrera y Jennyfer Mejía

Mediante Lambda se da la respuesta de Alexa al usuario para la cual se necesita los “*Samples*” y se evalúa los “*Slots*” obtenido de la Skill.

La comunicación de Lambda a un objeto IoT se realiza mediante el servidor MQTT por el puerto 8883 (Puerto estándar para comunicaciones sin encriptar, usan SSL/TLS), que proporcionan la comunicación de red segura mediante el protocolo

TLS, así como el uso de autenticación por medio de certificados proporcionados por AWS IoT. El mensaje antes de ser publicado pasa por un objeto de JavaScript y se convierte en una cadena codificada por JSON.

Para conexión con un servidor MQTT hay varios indicadores y parámetros que un cliente puede establecer como se observa en la Tabla 3.2:

Tabla 3.2. Funciones de MQTT.

MQTTConnect	Se utiliza para conectarse a un servidor MQTT (también llamado "Broker").
MQTTDisconnect	Le permite desconectarse de un servidor MQTT.
MQTTPublish	Publica un mensaje en el agente MQTT.
MQTTSubscribe	Se utiliza para suscribirse a una notificación MQTT.
MQTTUnsubscribe	Se emplea para cancelar la suscripción de una notificación MQTT.
MQTTUnsubscribeAll	Le permite darse de baja de todas las suscripciones existentes.

Indicadores y parámetros para una conexión con MQTT. Fuente: (SOFT, 2020)

La conexión a un servidor MQTT fue realizado mediante la función de *"mqttpublish"*, así publicará un mensaje en un servidor MQTT para lo cual se necesita conocer la dirección IP o el nombre del dispositivo y el puerto que está utilizando. Para el presente proyecto se nombró como *"aquarium"* el cual sirve como única conexión de cada función Lambda y habilidad ya que se utiliza la Raspberry Pi como único cliente y el puerto 8883.

### 3.7.1.2 Función AWS Lambda para el teclado

Para el aprendizaje del alfabeto y números en código braille por medio del teclado será activado por el comando de voz, con las palabras claves "Alexa iniciar teclado braille" para lo cual se creó la función llamada *"Teclado"* que cumple los mismos pasos de la función antes mencionada con la diferencia de los *"Intents"*, *"Samples"* y *"Slots"*.

El *"Intents TecfeedIntenet"* que llama a la función con el mismo nombre para enviar la respuesta de Alexa por medio la variable *"cardTitle"* que publica el mensaje y *"speechOutput"* que lo reproduce, *"Samples"* evalúa la lista de *"Slots"* en este caso la palabra teclado que será enviada por medio *"mqttPublish"* a la Raspberry Pi y ser interpretada.

### 3.7.2 Etapa de evaluación

En la etapa de evaluación se utilizaron las funciones de “*PruebaABC*” y “*PruebaNum*”, se tomará como ejemplo la primera función. Como se mencionó anteriormente la función cumple los pasos idénticos para la creación del código con la diferencia que se creó dos funciones, la primera realiza la pregunta en el caso de “*PruebaABC*” la letra y “*PruebaNum*” el número, la segunda función para la respuesta con los diferentes fonemas que puede interpretar la Skill Alexa.

La función “*pruebaabcbgioIntent*” para que Alexa realice la pregunta “Qué Letra es” para lo cual el usuario debe mencionar la palabra “Siguiete” para activar los solenoides con las diferentes letras de manera randómica, así mismo para la evaluación de los números. En la siguiente función “*abcfedIntent*” de respuesta donde se encuentran los diferentes fonemas para cada letra del abecedario que son evaluados por Lambda en una lista de “Slots” así por ejemplo el usuario al decir “jota” o “jota minúscula” será interpretado primero por de la skill de Alexa para ser enviada a Lambda y publicar a la Raspberry Pi por medio de MQTT.

## 3.8 Código Python

El software del dispositivo braille que se utilizo es Python dentro del sistema operativo de la Raspberry Pi, el cual consta de dos partes donde se describe las librerías y líneas de código más importantes tanto para el teclado braille y la activación de los solenoides.

### 3.8.1 Código Python para el Teclado braille

El programa para el uso del teclado matricial se realizó por medio de una matriz 5x8, en vectores con los diferentes caracteres mediante la librería “os” el cual permite acceder a funcionalidades dependientes del Sistema Operativo para ello se debe ser declarada en una la línea de código el reproductor como el archivo en mp3 como se observa en la Figura 3.16. Los archivos de audio mp3 se deben encontrar en una carpeta o la misma “ubicación que el código Python.

Figura 3.16. Librería os.system

```
if MATRIX[i][j] == "1":  
    os.system('omxplayer uno.mp3 &')  
    time.sleep(.3)
```

Librería os para reproducir un archivo formato mp3. Elaborado por: Paola Cabrera y Jennyfer Mejía.

Para la creación de los archivos de audio se realizó mediante gTTS que es una aplicación de Google para leer en voz alta el texto en varios idiomas, así por medio de un programa en Python convierte un texto deseado a voz por ejemplo vocales, consonantes, letras, números y frases en formato mp3, como se muestra en la Figura 3.17. Se escogió el sintetizador de voz Google porque se asemeja a la voz de Alexa para que los usuarios se familiaricen con ella.

Figura 3.17. Sintetizador de voz gTTS

```
from gtts import gTTS  
  
eng = "One"  
obj = gTTS(text = eng, lang = 'en')  
obj.save('eng.mp3')  
  
spanish = "uno"  
obj = gTTS(text = spanish, lang = 'es')  
obj.save('uno.mp3')
```

Ejemplo de la conversión de texto a voz. Elaborado por: Paola Cabrera y Jennyfer Mejía.

### 3.8.2 Código Python y MQTT

Para la comunicación con Alexa y Raspberry Pi tanto para el teclado como los solenoides se realizó en un mismo código Python, al igual que en el script de Lambda se debe tener en cuenta la ubicación de las credenciales proporcionadas por AWS IoT las cuales sirven para proteger la conexión, el nombre de “host”, “topic” y nombre de los certificados X.509, como se muestra en la Figura 3.18.

Figura 3.18. Configuración para comunicación a AWS IoT

```
.....  
cert_path = "/home/pi/Desktop/certs3/"  
host = "a2xbgqcxj4fvuf-ats.iot.us-east-1.amazonaws.com"  
topic = "$aws/things/aquarium/shadow/update"  
root_cert = cert_path + "AmazonRootCA1.key"  
cert_file = cert_path + "cert.pem.crt"  
key_file = cert_path + "private.pem.key"  
globalmessage = "" # to send status back to MQTT  
isConnected = False
```

Ubicación de certificados, host y credencial IoT. Elaborado por: Paola Cabrera y Jennyfer Mejía

El dispositivo necesita dos librerías principales las cuales son: “*JSON*” que facilita el intercambio de datos y, “*paho.mqtt.client*” para crear un objeto que sea cliente MQTT.

Para que el cliente se conecte a un agente MQTT necesita los parámetros de host, el puerto y vincular la dirección. La llamada a la función “*on\_connect*” la cual funciona cuando se pierde la conexión y se vuelve a conectar, las suscripciones se renovarán, la función “*on\_message*” que es la devolución de llamada para cuando se recibe un mensaje “*publish*” del servidor, en esta parte se define todos los “*Intents*” que se creó en las skills de Alexa mediante el parámetro *data* para enviar y recibir datos desde y hacia los componentes de AWS IoT, por último, el puerto por el cual se va a conectar.

A continuación, para la activación tanto del teclado braille como las salidas GPIO para los solenoides se debe llamar cada la función que contiene los “*Intent*” de Alexa visto anteriormente, así una vez realizada la comunicación se recibe el mensaje y el dato será evaluado para saber que Slots fue enviado por Lambda.

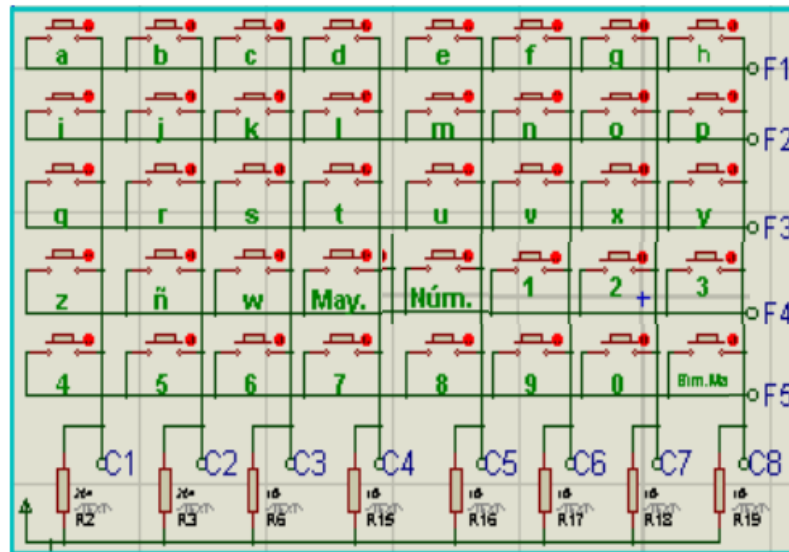
### **3.9 Implementación de hardware**

El diseño de los componentes electrónicos se divide en dos partes, el sistema de potencia y el sistema de control. El sistema de control es comandado por un Raspberry Pi, seguido de las impresiones de los circuitos finales y las teclas en impresión 3D.

#### **3.9.1 Implementación del teclado braille**

Para la realización del teclado braille se creó una matriz de 8x5 con pulsadores pull-up considerando el voltaje de 5V y una corriente de 3mA que proporciona la Raspberry como se puede observar en la Figura 3.19, se encuentra cada uno de los caracteres el alfabeto y números además del signo generador.

Figura 3.19. Teclado matricial 8x5



Teclado matricial etapa de aprendizaje. Elaborado por: Paola Cabrera y Jennyfer Mejía.

El valor obtenido de la resistencia es de  $2,2k[\Omega]$ , sin embargo, representa el límite de la capacidad de corriente que soporta cada pin la Raspberry, lo cual no es favorable por lo que se recomienda utilizar una resistencia de  $2,2[K\Omega]$  a  $10[K\Omega]$  con una tolerancia del 5% y una potencia  $\frac{1}{4}$  de Vatio, para trabajar de manera eficiente. El cálculo se realizó mediante la Ley de Ohm descrito a continuación:

$$V = R \times I \quad \text{Ec. (3.1)}$$

Donde  $V =$  Voltaje [V]  
 $R =$  Resistencia [ $\Omega$ ]  
 $I =$  Corriente [A]

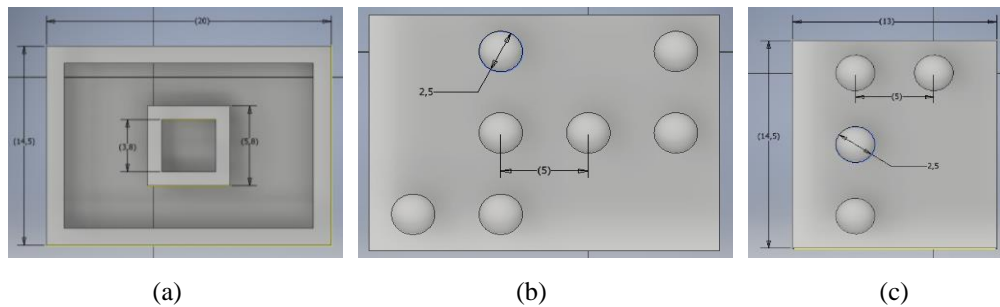
Despejando la Resistencia de la Ecuación 1:

$$R = \frac{V}{I} = \frac{5}{3m} = 1.666,667 \approx 2,2k\Omega$$

El modelo de las teclas se realizó mediante el Software Inventor como se observa en la Figura 3.20 ítem A, la tecla es diseñada para ajustarse al pulsador y así mantenerse fija, en el ítem B representa del número cero con las dimensiones del relieve y

distancia, por último, en el ítem C sirve como ejemplo de cómo se encuentra realizado las teclas del abecedario, con el relieve necesario así los usuarios podrán sentir con dedos para examinar o reconocer cada uno de los caracteres.

Figura 3.20. Diseño de las teclas en código braille



Diseño y cotas de las teclas en código braille. Elaborado por: Paola Cabrera y Jennyfer Mejía

### 3.9.2 Placa de protección

Los solenoides al tener un amperaje alto se necesita evitar el regreso de corrientes que puedan dañar la Raspberry para lo cual se desarrolló una placa con diodos 1N4007 los cuales soportan una corriente de hasta 1A de forma continua, además picos de voltaje inverso de hasta 1000v; estos diodos van conectados en antiparalelo a cada solenoide como se muestra en la Figura 3.21.

Figura 3.21. Placa de protección



Diodos para evitar el retorno de la corriente. Elaborado por: Paola Cabrera y Jennyfer Mejía

### 3.9.3 Diseño de la Estructura

El material del prototipo se encuentra realizado en tol galvanizado como se muestra en la Figura 3.22 ítem A, en el ítem B se observa la ubicación de los elementos electrónicos, la batería y los solenoides, gracias al programa fusión 360 de Autodesk da una vista más clara y precisa de cómo va quedar el diseño final.

Figura 3.22. Parte interna y externa del prototipo

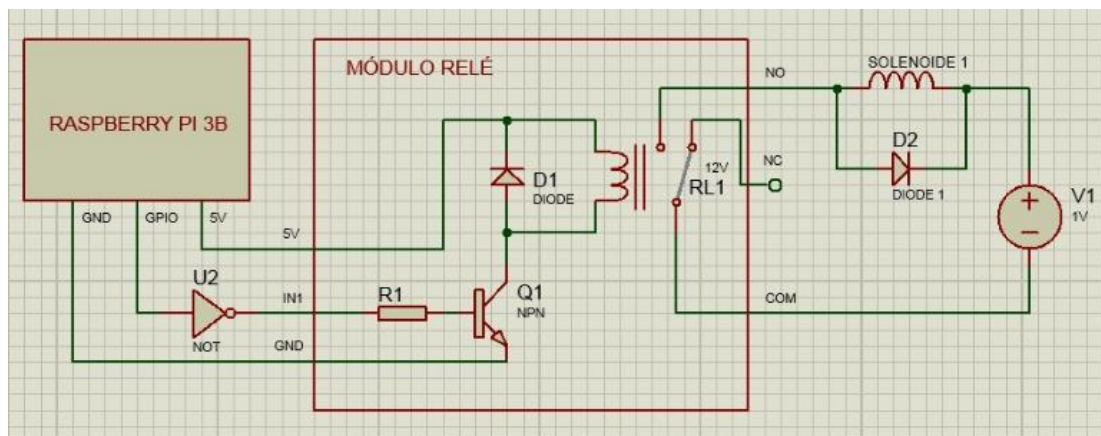


Vista de la estructura interna y externa de la estructura. Elaborado por: Paola Cabrera y Jennyfer Mejía.

### 3.9.4 Diseño electrónico

Para la activación de los solenoides se necesitó una fuente de 12v por lo cual en el circuito de control se utilizó un módulo relé para aislar el circuito de potencia. Este módulo relé, está energizado con 5v y es activado mediante las salidas de la Raspberry Pi como se muestra en la Figura 3.23.

Figura 3.23. Circuito electrónico



Esquema de conexión de un solenoide. Elaborado por: Paola Cabrera y Jennyfer Mejía.



## CAPÍTULO 4

### ANÁLISIS DE FUNCIONAMIENTO Y RESULTADO

En el presente capítulo se da a conocer el funcionamiento y resultados obtenidos del prototipo para ello se llevará a cabo diversos experimentos de reconocimiento de voz en condiciones ideales como una eficaz conexión a internet y sin interferencias externas, en 35 repeticiones para cada aplicación de las skills. Para capturar la señal de voz se utilizó una diadema con micrófono como se muestra en la Figura 4.1. A continuación, se describen las pruebas y los resultados obtenidos en cada una de ellas. Además, se utilizaron usuarios que tienen discapacidad visual y voluntarios quienes fueron vendados, los cuales realizaron pruebas sobre la funcionalidad del sistema.

Figura 4.1. Pruebas con la usuaria Anabel con discapacidad visual



Prueba de funcionamiento con la usuaria. Elaborado por: Paola Cabrera y Jennyfer Mejía.

#### 4.1.Pruebas de idioma

Alexa dispone de idiomas como en catalán, japonés, italiano, alemán, francés, inglés y español, los cuales para el idioma en español tenemos de México, España y de Estados Unidos dando este último la posibilidad de entender comandos en inglés y en español como se muestra en la Tabla 4.1. Se realizó las pruebas ya que inicialmente

se escogió por el español de España y al pronunciar las no había respuesta o confundía con otras letras como por ejemplo en la letra “P” se reconocía como “B” y la letra “F” por “S”, también había diferencia en la pronunciación de ciertas letras como “w” que en España se pronuncia “uve doble” y en México “doble u”.

Tabla 4.1. Pruebas con el idioma español de México, España y Estados Unidos.

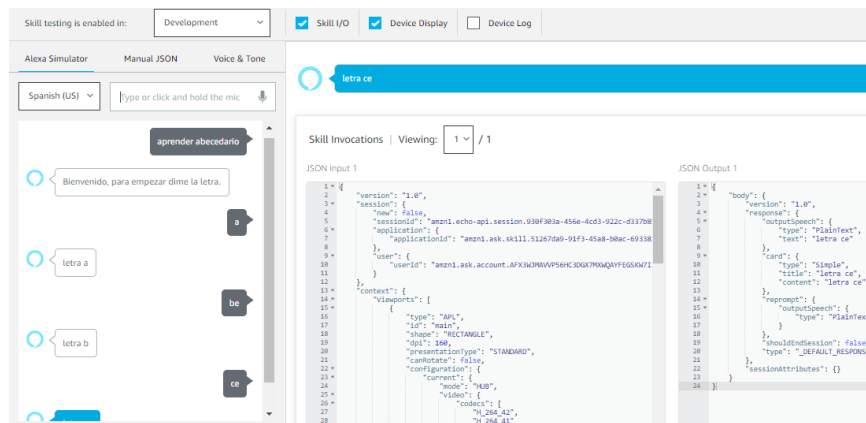
IDIOMA LETRAS	US	MEX	SPAIN
a	✓	✓	✓
be	✓	✓	✓
ce	✓	✓	✓
de	✓	✓	✓
e	✓	✓	✓
efe	✓	x	x
ge	✓	✓	✓
hache	✓	✓	✓
i	✓	x	x
jota	✓	✓	✓
ka/ca	✓	✓	✓
ele	✓	✓	✓
eme	✓	✓	✓
ene	✓	✓	✓
enie	✓	✓	✓
o	✓	✓	✓
pe	✓	x	✓
cu	✓	✓	✓
ere/erre	✓	✓	✓
ese	✓	✓	✓
te	✓	✓	✓
u	✓	✓	✓
uve	✓	✓	✓
doble ve/uve doble	✓	✓	✓
equis	✓	✓	✓
i griega	✓	✓	✓
zeta/ceta	✓	✓	✓

Prueba idioma español en diferentes países. Elaborado por: Paola Cabrera y Jennyfer Mejía

## 4.2. Pruebas de comunicación skill de Alexa a Lambda AWS

Para las pruebas de conectividad se realizó de dos formas, una durante la programación y desarrollo gracias a las secciones de Analytics y Testing que proporciona Alexa Developer Console, también se realiza mediante una prueba en la sección Test que es un simulador que da acceso a las funciones de la skill de Alexa interactuando con la voz o texto como se muestra en la Figura 4.2, para lo cual se escribe el nombre la skill “aprender abecedario”.

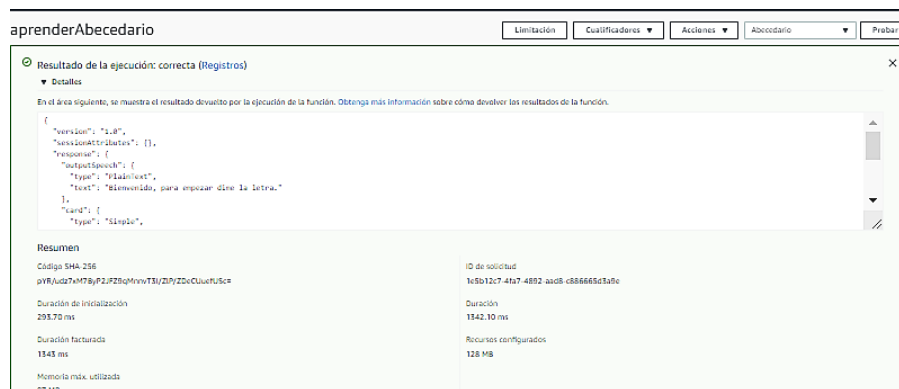
Figura 4.2. Prueba en la sección Test de la Alexa Developer Console



Prueba de invocación de la Skill aprender abecedario. Elaborado por: Paola Cabrera y Jennyfer Mejía.

La segunda forma es por medio de AWS Lambda mediante los eventos de prueba como se muestra en la Figura 4.3 que verifica el estado correctamente de las interfaces de Alexa y Lambda.

Figura 4.3. Estado de prueba entre la habilidad de Alexa y Lamba



Prueba desde AWS Lambda a la Skill aprender abecedario. Elaborado por: Paola Cabrera y Jennyfer Mejía

### 4.3. Comunicación a la Raspberry Pi

Para la conectividad o respuesta en el dispositivo se realiza de dos formas, una es mediante AWS IoT que permite la monitorización como se observa en la Figura 4.4 donde se aprecia el número de mensajes publicados y recibidos, el protocolo MQTT, las conexiones realizadas y el número de conexiones establecidas a lo largo del tiempo en un día de pruebas como resultado el pico más alto fue 268 conexiones realizadas correctamente.

Figura 4.4. Monitorización por medio de la plataforma AWS IoT.



Conexiones realizadas correctamente y mensaje recibidos. Elaborado por: Paola Cabrera y Jennyfer Mejía

La siguiente forma es por medio de la Raspberry Pi y el protocolo MQTT mediante "publicar-suscribir". Con pub-sub recibirá mensajes los cuales observa en el terminal del dispositivo cómo se muestra en la Figura 4.5. Una vez ingresado a la skill de aprendizaje de los números y al mencionar determinado dígito se recibirá el mensaje y llegará hasta el dispositivo para activar los solenoides y confirmando el mensaje en la pantalla de la tarjeta.

Es importante tener en cuenta que la respuesta del dispositivo dependerá de la calidad de conexión de internet y del tiempo de retraso en la comunicación entre el sistema con altavoz Echo y la Raspberry Pi, y los servidores de Amazon.

Figura 4.5. Mensajes recibidos por la Raspberry Pi

```
pi@raspberrypi:~/Desktop $ cd tesis_final_final
pi@raspberrypi:~/Desktop/tesis_final_final $ python tesis_completa.py
Gracias por usar mi programa. Cuidate Bye Bye!
^Cpi@raspberrypi:~/Desktop/tesis_final_final $ python tesis_completa.py
Numero tres
Numero cuatro
Numero cinco
Numero seis
Numero siete
Numero ocho
Numero nueve
Numero cero
Letra f
```

Mensajes correctamente recibidos. Elaborado por: Paola Cabrera y Jennyfer Mejía

En la Figura 4.6. se muestra las pruebas realizadas en la evaluación de los números cada vez que el usuario mencione el comando “siguiente” a Alexa se enviará un mensaje “¿Qué número es?” a la tarjeta y cuando el usuario responda recibirá un mensaje con la validación de “correcto” e “incorrecto”. La programación tanto de aprendizaje, evaluación y el teclado se encuentran realizadas en un mismo script.

Figura 4.6. Mensajes recibidos de la evaluación de los números

```
Que numero es
Correcto cinco
Audio codec mp3float channels 2 samplerate 48000 bitspersample 16
Subtitle count: 0, state: off, index: 1, delay: 0
have a nice day ;)
Que numero es
Correcto uno
Audio codec mp3float channels 2 samplerate 48000 bitspersample 16
Subtitle count: 0, state: off, index: 1, delay: 0
have a nice day ;)
Que numero es
Correcto cuatro
Audio codec mp3float channels 2 samplerate 48000 bitspersample 16
Subtitle count: 0, state: off, index: 1, delay: 0
have a nice day ;)
Que numero es
Correcto dos
Audio codec mp3float channels 2 samplerate 48000 bitspersample 16
Subtitle count: 0, state: off, index: 1, delay: 0
have a nice day ;)
Que numero es
Correcto tres
```

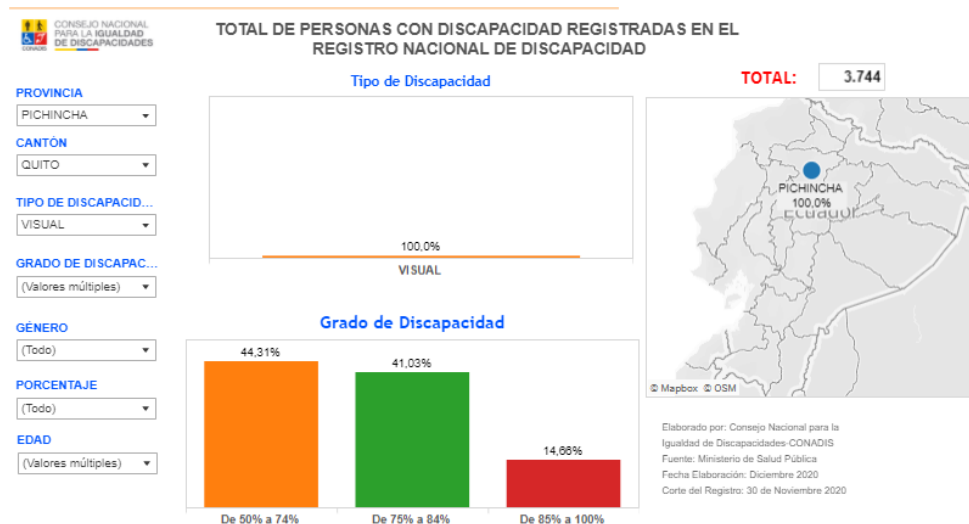
Mensajes recibidos de la script de evaluación números. Elaborado por: Paola Cabrera y Jennyfer Mejía

#### 4.4.Pruebas con el usuario

Para la realización de las pruebas con los usuarios se tomó como muestra un número de 35 repeticiones determinada por la ecuación 4.1, con una población de 3744 personas con discapacidad visual según los datos registrados en el registro nacional de

discapacidad, se tomó en cuenta la provincia de Pichincha, en el cantón Quito, personas con un grado de discapacidad entre el 50% al 100% y edades entre 7 a 65 años como se muestra en la Figura 4.7.

Figura 4.7. Estadísticas de Discapacidad visual en la Provincia de Pichincha



Total, de personas con discapacidad visual registradas en la provincia de Pichincha. Elaborado por: Paola Cabrera y Jennyfer Mejía.

$$n = \frac{k^2 N p q}{e^2 (N - 1) + k^2 p q} \quad \text{Ec. (4.1)}$$

$$n = \frac{1.65^2 * 3744 * 0.9 * 0.1}{0.10^2 (3744 - 1) + 1.65^2 * 0.9 * 0.1}$$

$$n = 35$$

### Prueba 1.- Invocación principales de skills de Alexa

Al realizar las pruebas con las invocaciones de las cinco principales skills de Alexa se obtuvo un porcentaje de reconocimiento del 92.00%, como se aprecia en la Tabla 4.2, debido a que al llamar la skill se producía errores especialmente en “aprender abecedario” ya que confundía con otras habilidades y “teclado braille” porque al no cerrar la skill con el comando concluir en el siguiente llamado da un error.

Tabla 4.2. Muestras tomadas de las principales habilidades de Alexa

Comando	Nº de muestras	Acierto	Desacierto	% de reconocimiento
Aprender abecedario	35	33	2	94,29
Aprender números	35	32	3	91,43
Teclado braille	35	31	4	88,57
Evaluación abecedario	35	33	2	94,29
Evaluación números	35	32	3	91,43
PROMEDIO				92,00

Muestras tomadas de las principales habilidades con sus porcentajes de reconocimiento válidos.

Elaborado por: Paola Cabrera y Jennyfer Mejía.

El porcentaje de reconocimiento se calculó mediante la ecuación 4.2.

$$\% \text{ de reconocimiento} = \frac{n^{\circ} \text{ aciertos} * 100}{n^{\circ} \text{ muestras}} \quad \text{Ec. (4.2)}$$

## Prueba 2.- Skill Aprender números

Al realizar las pruebas con la skill de aprendizaje para los números se obtuvo un porcentaje mayor del 99,71% como se indica en la Tabla 4.3, debido a que se utilizó palabras y no sílabas, con la excepción del número uno que obtuvo un error un bajo ya que al iniciar con la habilidad no registraba por ser la primera palabra en reconocer.

Tabla 4.3 Muestras tomadas en la Skill aprender números

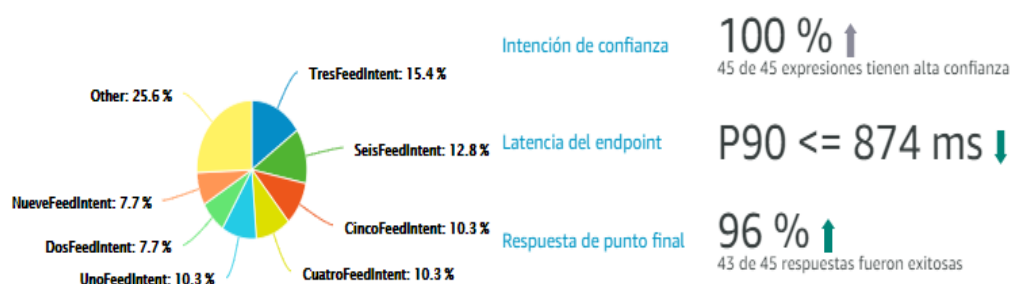
Comando	Nº de muestras	Acierto	Desacierto	% de reconocimiento
uno	35	34	1	97,14
dos	35	35	0	100,00
tres	35	35	0	100,00
cuatro	35	35	0	100,00
cinco	35	35	0	100,00
seis	35	35	0	100,00
siete	35	35	0	100,00
ocho	35	35	0	100,00
nueve	35	35	0	100,00
cero	35	35	0	100,00
PROMEDIO				99,71

Muestras tomadas de la Skill de aprendizaje aprender números con sus porcentajes de reconocimiento

válido. Elaborado por: Paola Cabrera y Jennyfer Mejía.

En la Figura 4.8 se muestra los porcentajes de los llamados de la skill de los números donde se puede medir las veces que se menciona un determinado número, gracias a la consola de desarrollo de Alexa ingresando en la pestaña “Analítica” se puede medir el rendimiento de la skill mostrando la latencia y el porcentaje de respuesta que en este caso fue de 874ms en el tiempo de respuesta y 96% de respuestas exitosas.

Figura 4.8. Porcentaje de veces que se mencionó un número y porcentaje de fiabilidad de la skill.



Porcentajes de veces que se llama a cada número, latencia y porcentajes de respuesta como confianza. Elaborado por: Paola Cabrera y Jennyfer Mejía.

### Prueba 3.- Skill Aprender abecedario

Para la skill de aprendizaje del abecedario se obtuvo un 98.20% de efectividad como se muestra en la Tabla 4.4, ya que algunas letras como en el caso de la “a” al iniciar la Skill no reconocía correctamente, en otras letras tales como la c, d, o, p, u y w confundía con otras letras de acento similar como por ejemplo la “p” reconocía en ocasiones como “g” o “t” para la cual se realizó pruebas como el añadir sinónimos a cada una de las letras que tenían error como por ejemplo para la letra c se incluyó sinónimos como: CE, CEE y CÉ, como se muestra en la Figura 4.9.

Figura 4.9. Sinónimos para el llamado de la letra C

VALUE ⓘ	ID (OPTIONAL) ⓘ	SYNONYMS (OPTIONAL) ⓘ
c	Enter ID	<input type="text" value="Add synonym"/> + <div> <span>CEE x</span> <span>CÉ x</span> <span>CE x</span> </div>

Corrección del error en el llamado de la letra C. Elaborado por: Paola Cabrera y Jennyfer Mejía

Además, se considera que la skill se ejecutó de una manera correcta ya que las letras tales como las “j”, “h” e “y” podrían ser confundidas con alguna otra letra debido por su pronunciación, pero se obtuvo una calificación ideal del 100%.



Tabla 4.4. Toma de muestras de la skill aprender abecedario

Comando	Nº de muestras	Acierto	Desacierto	% de reconocimiento
A	35	34	1	97,14
Be	35	35	0	100,00
Ce	35	31	4	88,57
De	35	32	3	91,43
E	35	35	0	100,00
Efe	35	35	0	100,00
Ge	35	35	0	100,00
Hache	35	35	0	100,00
I	35	35	0	100,00
Jota	35	35	0	100,00
ka/ca	35	35	0	100,00
Ele	35	35	0	100,00
Eme	35	35	0	100,00
Ene	35	35	0	100,00
Enie	35	35	0	100,00
O	35	33	2	94,29
Pe	35	33	2	94,29
Cu	35	35	0	100,00
ere/erre	35	35	0	100,00
Ese	35	35	0	100,00
Te	35	35	0	100,00
U	35	31	4	88,57
Uve	35	35	0	100,00
doble ve/uve doble	35	34	1	97,14
Equis	35	35	0	100,00
i griega	35	35	0	100,00
zeta/ceta	35	35	0	100,00
PROMEDIO				98,20

Muestras tomadas de la skill de aprender abecedario. Elaborado por: Paola Cabrera y Jennyfer Mejía.

En la Figura 4.10 se observa una latencia de 3674,1ms de tiempo de respuesta de la skill siendo 49 exitosas de 50 intentos, en la parte derecha también se puede observar el porcentaje de veces que se menciona una letra, se realizó la mayoría de pruebas con las letras que tenían tendencia a confundirse con otras.

Figura 4.10. Porcentajes de respuesta de la skill aprender abecedario



Porcentaje de veces que se llama a cada letra, latencia y porcentajes de respuesta. Elaborado por: Paola Cabrera y Jennyfer Mejía.

#### Prueba 4. Skill evaluación números

Para las pruebas realizadas con la skill de evaluación de números se obtuvo un porcentaje de reconocimiento del 100%, ya que al evaluar números aleatorios Alexa podía identificar cada uno de ellos sin ningún problema como se muestra en la Tabla 4.5.

Tabla 4.5. Toma de muestras de la skill evaluación números

Comando	Nº de muestras	Acierto	Desacierto	% de reconocimiento
uno	35	35	0	100,00
dos	35	35	0	100,00
tres	35	35	0	100,00
cuatro	35	35	0	100,00
cinco	35	35	0	100,00
seis	35	35	0	100,00
siete	35	35	0	100,00
ocho	35	35	0	100,00
nueve	35	35	0	100,00
cero	35	35	0	100,00
PROMEDIO				100,00

Muestras tomadas de la Skill de evaluación números con sus porcentajes de reconocimiento válido.

Elaborado por: Paola Cabrera y Jennyfer Mejía.

En la Figura 4.11 se muestra el porcentaje de confianza de un 100% siendo estas 58 respuestas exitosas de 61 intents con un tiempo de latencia de 1248,4ms, en la Figura ubicada en la parte derecha se muestra la estadística de invocación y siento estas “PruebaGpioIntenet” para las preguntas y “UnopFeedIntent” para la respuesta.

Figura 4.11. Datos estadísticos y comportamiento de la skill evaluación números



Confianza, latencia, respuestas exitosas y diagrama estadístico válido. Elaborado por: Paola Cabrera y Jennyfer Mejía.

### Prueba 5.- Skill evaluación abecedario

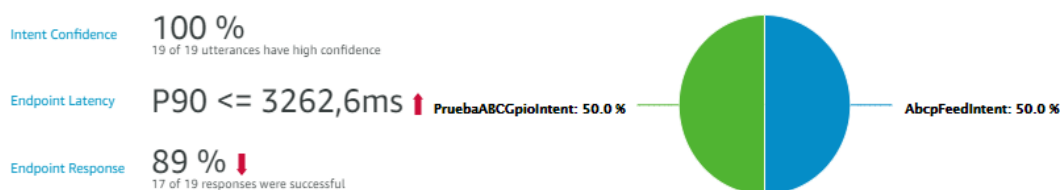
En la skill para la evaluación del abecedario se obtuvo un porcentaje de reconocimiento de 98,94% como se indica en la Tabla 4.6, en esta habilidad se obtuvo menor error en las letras como la c, d, o, p, u y w puesto que se corrigió en la skill de aprendizaje.

Como se puede observar la letra “c” tiene un porcentaje de error mayor tanto en la skill de aprendizaje como de evaluación, ya que es propensa a confundirse con otras letras del alfabeto tales como “e”, “de” y “ge” dependiendo de la velocidad con la que esta se pronuncia la letra antes mencionada.

De igual manera existe un porcentaje de error con la letra “u”, ya que se puede confundir con la letra “q”, para la corrección de dicho error se logró mediante pruebas con las distintas formas de pronunciar dicha letra como “ú” y “uu” y para la letra “q” se diferenció con los fonemas “ku” y “cu”.

Al igual a la skill de evaluación números se obtuvo un porcentaje de reconocimiento aceptable puesto que las letras que daban fallas fueron corregidas en la skill de aprendizaje, el tiempo de respuesta es de 3262,6ms ya que se debe evaluar a que letra pertenece desde la Raspberry Pi, dando como resultado 17 respuestas exitosas de 19. En la Figura ubicada en la parte izquierda nos muestra que el número de preguntas “PruebaABCGpioIntent” es igual al número de respuestas “AbcpFeedIntent” como se muestra en la Figura 4.12.

Figura 4.12. Datos estadísticos y diagrama de la Skill evaluación abecedario.



Confianza, latencia, respuestas exitosas y diagrama estadístico válido. Elaborado por: Paola Cabrera y Jennyfer Mejía.

Tabla 4.6. Toma de muestras de la skill evaluación abecedario.

Comando	Nº de muestras	Acierto	Desacierto	% de reconocimiento
A	35	34	1	97,14
Be	35	35	0	100,00
Ce	35	31	4	88,57
De	35	33	2	94,29
E	35	35	0	100,00
Efe	35	35	0	100,00
Ge	35	35	0	100,00
Hache	35	35	0	100,00
I	35	35	0	100,00
Jota	35	35	0	100,00
ka/ca	35	35	0	100,00
Ele	35	35	0	100,00
Eme	35	35	0	100,00
Ene	35	35	0	100,00
Enie	35	35	0	100,00
O	35	34	1	97,14
Pe	35	33	2	94,29
Cu	35	35	0	100,00
ere/erre	35	35	0	100,00
Ese	35	35	0	100,00
Te	35	35	0	100,00
U	35	33	2	94,29
Uve	35	35	0	100,00
doble ve/uve doble	35	33	2	94,29
Equis	35	35	0	100,00
i griega	35	35	0	100,00
zeta/ceta	35	35	0	100,00
PROMEDIO				98,94%

Porcentajes de reconocimiento válido de la skill. Elaborado por: Paola Cabrera y Jennyfer Mejía.

#### **4.5. Validación del prototipo**

La validación del prototipo se realizó con personas videntes y no videntes mediante una encuesta. Las preguntas se realizaron para respuestas de “sí” y “no” y son las siguientes:

##### **Pregunta 1.-**

¿Conoces alguna institución que ofrezca cursos de aprendizaje de lenguaje braille gratuitos o de bajo costo?

##### **Pregunta 2.-**

¿Estaría usted dispuesto a apoyar económicamente en este tipo de proyectos con el objetivo de que ciertas fundaciones con fines de inclusividad obtengan el dispositivo de manera totalmente gratuita?

##### **Pregunta 3.-**

¿Cree usted que los asistentes virtuales como ente caso Alexa de Amazon facilita a los no videntes la realización de tareas?

##### **Pregunta 4.-**

¿Cree usted que los centros de enseñanza para personas no videntes podrían hacer uso de este dispositivo y lograr un aprendizaje del sistema braille grado 1 en menor tiempo?

##### **Pregunta 5.-**

¿Le parece a usted que el prototipo realizado tiene un método de enseñanza totalmente autónomo?

##### **Pregunta 6.-**

En caso de tener un conocido no vidente, ¿recomendaría este dispositivo de enseñanza?

La encuesta realizada a 35 personas en edades de 30 a 50 años, según los resultados de la encuesta y haciendo uso de la ecuación 4.2 el proyecto tiene un promedio de aprobación del 90.17% como se muestra en la Tabla 4.7, quienes opinan que el proyecto es un recurso de gran apoyo para las personas no videntes, siendo un aprendizaje totalmente autónomo, además que teniendo el apoyo económico para poder incorporarlo en centros de apoyo o fundaciones su uso no tendría ningún valor. Adicional, al tener un asistente de voz como Alexa, resulta un aprendizaje muy amigable con el usuario, y acompañado de la etapa de evaluación permite que la

persona agilite el proceso de aprendizaje.

Tabla 4.7. Porcentaje de validación del proyecto.

PREGUNTA	PORCENTAJE DE VALIDACIÓN
1	83
2	77
3	95
4	92
5	96
6	98
PROMEDIO	90.17%

Tabulación sobre la encuesta realizada media en porcentajes para su validación. Elaborado por: Paola Cabrera y Jennyfer Mejía.

#### 4.6. Costos de implementación del dispositivo.

En la tabla 4.8 se observa los diferentes elementos utilizados para la elaboración del dispositivo para aprendizaje braille.

Tabla 4.8. Tabla de costos

Elemento	Cantidad	Valor unitario	Total
Raspberry Pi 3B	2	\$60	\$120
Solenoido electromagnético	10	\$6	\$60
Modulo relé de 16 canales	1	\$18	\$18
Fuente 24v	1	\$5	\$5
Teclas braille	49	\$0.5	\$24.5
Placa teclado	1	\$40	\$40
Placa protección	1	\$8	\$8
Placa conexión solenoides	1	\$9	\$9
Estructura en tol galvanizado	1	\$55	\$55
Parlante	2	\$5	\$10
Micrófono	1	\$7	\$7
TOTAL			\$ 356.5

Costos de implementación del dispositivo para aprendizaje braille. Elaborado por: Paola Cabrera y Jennyfer Mejía.

#### 4.7.Comparación de precios con dispositivos comerciales

En la Tabla 4.9 se muestra los dispositivos de aprendizaje del sistema braille comerciales tomando en cuenta las características que presenta el dispositivo diseñado en el presente trabajo ya que existen varios dispositivos electrónicos para la enseñanza del lenguaje braille difíciles de adquirir por los elevados costos.

Tabla 4.9. Comparación de costos con dispositivos similares

Nombre	Empresa	Bluetooth	USB	Asistencia De Voz	Audio	Wifi	Costo
Smart Beetle	HIMS	X	X				\$994.99
Ferrotouch		X	X		X	X	\$1000
Perkins Smart brailler	MAGNIFYI NG AIDS		X		X		\$2,695.00
Braibook	Braibook		X		X	X	\$500

Comparación de costos con dispositivos que cumplen con un funcionamiento similar. Elaborado por:

Paola Cabrera y Jennyfer Mejía.

## **CAPÍTULO 5**

### **CONCLUSIONES**

En base a la información recopilada de papers, documentos y prototipos existentes acerca del aprendizaje braille, se determinaron los componentes más adecuados para el prototipo, y en base a las características de la Tabla 2.1, se eligió a la tarjeta Raspberry Pi 3 B como la opción más óptima para el sistema sobre las demás con la finalidad de aprender y evaluar de una manera más dinámica, por su procesador y velocidad, así también como un asistente de voz que pueda comunicarse con la tarjeta de desarrollo, además de activar los actuadores; a diferencia de la tarjeta de desarrollo arduino que presenta menor capacidad de procesamiento y puertos de propósito general.

Con la información investigada se eligió como asistente de voz a Alexa ya que el sistema basado en la nube de Amazon, permite a la tarjeta Raspberry Pi funcionar como un servicio de voz, siendo este amigable con los usuarios permitiéndole interactuar con la tecnología de una manera natural gracias a su voz, la cual puede reconocer distintas voces de una manera sencilla. Gracias a sus características y servicios en la nube la comunicación con el dispositivo para la enseñanza del Sistema braille se realizó fácilmente, la cual se puede evidenciar en la Tabla 4.2 con un porcentaje de error del 8% que se considera ideal.

Debido a que el dispositivo fue utilizado por personas con discapacidad visual que inician un aprendizaje, se optó por el diseño de un teclado mediante una impresora en 3D para la elaboración de la simbología del alfabeto y números, de una manera proporcional al tamaño real de la celda braille, así el usuario al presionar la tecla pudo palpar el símbolo y este reproducía un audio que identificó dicho carácter, con la ayuda del asistente de voz se facilitó el aprendizaje siendo este más interactivo y autónomo.

La retroalimentación se realizó mediante solenoides, para la evaluación de lo aprendido tanto en el teclado braille como en la celdilla del signo generador, conformado por los actuadores, se creó una prueba randómica de números o letras tanto mayúsculas como minúsculas, mediante el asistente de voz y la tarjeta de



desarrollo, así el usuario una vez realizada su respuesta sería calificado con las palabras “Correcto” e “Incorrecto”, verificando lo aprendido con el dispositivo.

Se utilizó el protocolo MQTT (Message Queue Telemetry Transport) para realizar el sistema de retroalimentación, ya que este permite realizar aplicaciones con dispositivos IoT, así el usuario desde el asistente de voz envía un mensaje a la tarjeta Raspberry Pi y la placa de desarrollo a su vez identifica el mensaje y lo procesa para activar los actuadores en este caso los solenoides y el teclado.

La validación del funcionamiento del prototipo y el aprendizaje del lenguaje braille se realizó mediante pruebas de los distintos comandos de voz utilizado en cada una de la Skills, con ayuda de distintos tipos de voluntarios, es decir personas con discapacidad visual y sin ella, los cuales probaron el dispositivo, obteniendo un porcentaje de validación del 92% en las pruebas realizadas, como se muestra en la Tabla 4.2 por lo que se considera que el prototipo cumple con las expectativas realizadas. También se puede evidenciar en la Tabla 4.7 con la encuesta realiza dando un porcentaje de validación del 90,17%.

Este dispositivo es un nexo importante para la inclusión de personas con discapacidad visual, para el aprendizaje del sistema braille, como medio para la comunicación entre personas sin ninguna discapacidad y personas no videntes, ya que este dispositivo es una herramienta para todos los usuarios que deseen aprender e inicien una educación empezando por identificar el alfabeto y números en la simbología braille.

## RECOMENDACIONES

Al trabajar con el asistente de voz Alexa si se utiliza palabras o frases facilita el reconocimiento de voz evitando confundir con otros comandos a diferencia de utilizar sílabas como por ejemplo el comando “ce”, al igual que se debe tener en cuenta el idioma tanto del dispositivo *Amazon echo* como la *Skill de Alexa* debe ser el mismo, por último, tener en cuenta la posición correcta del micrófono si se utiliza una diadema y la tarjeta de desarrollo Raspberry Pi 3.

Al momento de trabajar con solenoides es ideal no utilizar por un período largo de tiempo, aproximadamente entre 45 minutos a una hora, especialmente cuando existe un error en el reconocimiento de voz y los solenoides se quedan enclavados, ya que produce un sobrecalentamiento de estas provocando un mal funcionamiento, consumiendo una excesiva corriente para lo cual también se sugiere un sistema de enfriamiento.

Con respecto a la tarjeta de desarrollo Raspberry Pi se propone una placa de protección constituida por diodos para evitar el regreso de corrientes, que puedan dañar el dispositivo al activar los solenoides. En la programación se sugiere tener en cuenta los tiempos del audio ya que tiempos muy cortos no se ejecutan con el reproductor “omxplayer”.

## REFERENCIAS

- Anoja Rajalakshmi, Hamid Shahnasser. (2017). Internet of Things using Node-Red and Alexa. *17th International Symposium on Communications and Information Technologies (ISCIT)*.
- Castaneda, S. (2014). *Sistema para el aprendizaje de lectoescritura en Braille "ABSe"*. Bogota.
- Castañeda, S. (2015). *ABSè Sistema para el aprendizaje de lectoescritura en Braille*. Bogotá: Pontificia Universidad Javeriana.
- Castilla, C. A. (2015). *Comunicación y Diversidad*. Bogota: Kimpres Ltda.
- Castillo, Á. A. (2019). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MQTT SIN BRÓKER BASADO EN SDN*. Granada: Universidad de Granada.
- COMISIÓN BRAILLE ESPAÑOLA, O. N. (2016). *ESCRITURA CON LA FUENTE BRAILLE DE LA COMISIÓN BRAILLE ESPAÑOLA*. Madrid: Departamento de Promoción Cultural y Braille. ONCE. Dirección General.
- CONADIS. (Julio de 2020). *Estadísticas de Discapacidad*. Obtenido de Consejo Nacional para la Igualdad de Discapacidades: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>
- Corso, L. D. (2016). *Polipropileno*.
- Developer, A. (2010-2021). *ASK SDK for Node.js*. Obtenido de Amazon Alexa Skill: <https://developer.amazon.com/en-US/docs/alexa/alexa-skills-kit-sdk-for-nodejs/overview.html>
- Diamantaras, I. (2019). *Interfaz de usuario de voz para sistemas de Automatización*. Alicante: Universidad de Alicante, Master Universitario en Automática y Robótica.
- Discapacidades, S. T. (2019). *Vicepresidencia de la República del Ecuador*. Obtenido de <https://www.vicepresidencia.gob.ec/secretaria-tecnica-de-discapacidades/#>
- Escobar, C. P. (2017). Implementación de un prototipo de impresora braille de bajo costo, basado en. *Revista Publicando*, 89-107.
- Escobar, C. P. (2017). Implementación de un prototipo de impresora braille de bajo costo, basado en hardware libre. *Revista Publicando*, 91.
- FERRARO, H. (2015). *RECONOCIMIENTO AUTOMATICO DE TEXTO BRAILLE. LA PLATA - ARGENTINA : FACULTAD DE INFORMATICA UNPL*.
- Flore, G. R. (2017). *Equation Chapter 1 Section 1 Trabajo Fin de Carrera Grado en*

*Ingeniería de las Tecnologías de Telecomunicación* Desarrollo de una aplicación web con Node.js para la monitorización en tiempo real de un electrocardiograma. Sevilla: Departamento de Electrónica Escuela Técnica Superior de Ingeniería Universidad de Sevilla.

GAIBOR, M. J. (2016). *ACCESO LABORAL DE LAS PERSONAS CON DISCAPACIDAD VISUAL EN EL DISTRITO METROPOLITANO DE QUITO: INDICADORES Y ESTRATEGÍAS QUE FOMENTEN SU INCLUSIÓN SOCIAL*. Quito: PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

GAIBOR, M. J. (2016). *ACCESO LABORAL DE LAS PERSONAS CON DISCAPACIDAD VISUAL EN EL DISTRITO METROPOLITANO DE QUITO: INDICADORES Y ESTRATEGÍAS QUE FOMENTEN SU INCLUSIÓN SOCIAL*. Quito: PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

GONZAGA, M. J. (2015). *IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA EL MONITOREO DE ACTIVIDADES REALIZADAS EN EL INSTITUTO DE CIENCIAS BÁSICAS, A TRAVÉS DE UN DISPOSITIVO MÓVIL*. Quito: Universidad Central del Ecuador.

Guzmán, T. B. (2016). *PRENDIENDO BRAILLE JUNTO A CANTALETRAS*. Chile: Especialista en Educación Diferencial, con Especialidad en Trastornos de la Visión, Universidad de Chile.

Haber-Olguin, A. (2015). Dispositivo electrónico simulador del código Braille con interfase multimedia interactiva. *Medigraphic Artesisa en línea*, 32-36.

Hernández, C. (2016). Dispositivo tecnológico para la optimización del tiempo de aprendizaje del lenguaje Braille en personas invidentes. *REVISTA DE SALUD PÚBLICA*, 865-873.

HIDALGO, J. C. (2018). *DISEÑO Y DESARROLLO DE UN PROTOTIPO DE LÍNEA BRAILLE DEBAJO COSTO PARA PERSONAS NO VIDENTES EN EL MARCO DE CÁTEDRA UNESCO "TECNOLOGÍAS DE APOYO PARA LA INCLUSIÓN EDUCATIVA" DE LA UNIVERSIDAD POLITÉCNICA SALESIANA*. Cuenca: Repositorio UPS.

Kaundinya, A. S. (2017). Voice Enabled Home Automation Using Amazon Echo. *International Research Journal of Engineering and Technology (IRJET)*, 682-684.

- MARTÍNEZ-LIÉBANA, I. (2015). *GUÍA DIDÁCTICA PARA LA LECTOESCRITURA BRAILLE*. España: Organización Nacional de Ciegos Españoles (ONCE).
- Mejía, C. A. (2019). *Diseño e implementación de un dispositivo electrónico interactivo para aprendizaje de Pre-Braille*. Obtenido de <http://201.159.222.99/bitstream/datos/8920/1/14568.pdf>
- MIRANDA, C. A. (2018). *Desarrollo de un prototipo de lectoescritura asistido por audio para el aprendizaje de personas no videntes mediante el sistema braille*. Obtenido de Repositorio Institucional de la Universidad Politécnica Salesiana / Tesis / Grado: <https://dspace.ups.edu.ec/bitstream/123456789/16164/1/UPS-ST003804.pdf>
- MIRANDA, C. A. (2018). *Desarrollo de un prototipo de lectoescritura asistido por audio para el aprendizaje de personas no videntes mediante el sistema braille*. Quito - Ecuador: Universidad Politecnica Salesiana. Obtenido de Repositorio Institucional de la Universidad Politécnica Salesiana / Tesis / Grado: <https://dspace.ups.edu.ec/bitstream/123456789/16164/1/UPS-ST003804.pdf>
- Mora, S. L. (2016). *Teclados Braille*. Obtenido de accesibilidadweb: <http://accesibilidadweb.dlsi.ua.es/?menu=tec-braille>
- Moya, P. A. (2017). *ELABORACIÓN DE UN JUGUETE INTERACTIVO SONORO PARA LA ENSEÑANZA DEL SISTEMA BRAILLE*. Quito: Universidad de las Américas.
- Ms.R.Rajyashree, A. P. (2018). Automation Using Alexa and Raspberry Pi. *International Research Journal of Engineering and Technology (IRJET)*, 828-831.
- Otero, C. (08 de Noviembre de 2018). *Betech*. Obtenido de Alexa de Amazon: cómo funciona el asistente, qué hace y los modelos Echo que hay: [https://as.com/meristation/2018/10/31/betech/1541025737\\_394177.html](https://as.com/meristation/2018/10/31/betech/1541025737_394177.html)
- Pastor, J. (25 de Abril de 2018). *Raspberry Pi 3 Model B+, análisis: más potencia y mejor WiFi para un miniPC que sigue asombrando*. Obtenido de Xataka: <https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando>
- Pauta, J. A. (2016). *Prototipo Electrónico de Enseñanza del Sistem Braille*. Cuenca.
- Peñaloza, O. P. (2006). Sistema electrónico braille para la ayuda en el aprendizaje de personas no videntes. *Creative Commons*, 209-225.

- Riazi, A. (Junio de 2016). *PMC*. Obtenido de [www.ncbi.nlm.nih.gov/pmc/articles/PMC4909643/](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4909643/)
- Rivera, C. F. (Oct de 2016). *Repositorio Universidad Técnica de Ambato*. Obtenido de <http://repositorio.uta.edu.ec/handle/123456789/24019>
- Rouse, M. (Mayo de 2019). *Searchaws Techtarget*. Obtenido de Amazon EC2 (Elastic Compute Cloud) : <https://searchaws.techtarget.com/definition/Amazon-Elastic-Compute-Cloud-Amazon-EC2>
- Sánchez, A. A. (feb de 2016). *Repositorio Universidad Técnica de Ambato*. Ambato: Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Electrónica y Comunicaciones. Obtenido de <http://repositorio.uta.edu.ec/jspui/handle/123456789/20344>
- Sánchez, A. A. (feb de 2016). *Repositorio Universidad Técnica de Ambato* . Obtenido de <http://repositorio.uta.edu.ec/jspui/handle/123456789/20344>
- Sánchez, A. A. (2016). *SISTEMA ELECTRÓNICO PARA LA ENSEÑANZA DEL LENGUAJE BRAILLE A PERSONAS INVIDENTES*. Ambato.
- Simón, C. (1995). El sistema Braille: Bases para su enseñanza-aprendizaje. *Comunicación, Lenguaje y Edoinción*(28), 91-102.
- SOFT, P. (04 de Abril de 2020). *MQTT functions*. Obtenido de <https://help.windev.com/en-US/?1000023078>
- Tapashetti, A. (2016). *IoT-Enabled Air Quality Monitoring Device*. Seattle, WA, USA: IEEE.
- TITUANA, M. L. (2016). *IMPACTO DE LAS APLICACIONES MULTIMEDIA EN EL PROCESO ENSEÑANZA/APRENDIZAJE DE LOS ESTUDIANTES CON DISCAPACIDAD VISUAL*. Obtenido de REPOSITORIO UNIVERSIDAD TÉCNICA DE MACHALA: <http://repositorio.utmachala.edu.ec/bitstream/48000/7551/1/TESIS%20COMPLETA.pdf>
- TITUANA, M. L. (2016). *IMPACTO DE LAS APLICACIONES MULTIMEDIA EN EL PROCESO ENSEÑANZA/APRENDIZAJE DE LOS ESTUDIANTES CON DISCAPACIDAD VISUAL*. Machala. Obtenido de REPOSITORIO UNIVERSIDAD TÉCNICA DE MACHALA: <http://repositorio.utmachala.edu.ec/bitstream/48000/7551/1/TESIS%20COMPLETA.pdf>

Yue, C. Z. (2017). Voice Activated Smart Home Design and Implementation. *2nd International Conference on Frontiers of Sensors Technologies* (págs. 489-492). Singapore: IEEE.